International Journal of Computer Vision 41(1/2), 109–134, 2001

The Joy of Sampling

D.A. FORSYTH, J. HADDON AND S. IOFFE
Computer Science Division, University of California at Berkeley, Berkeley, CA 94720, USA
daf@cs.berkeley.edu
haddon@cs.berkeley.edu
ioffe@cs.berkeley.edu

Received November 12, 1999; Revised August 4, 2000; Accepted January 17, 2001

Abstract. A standard method for handling Bayesian models is to use Markov chain Monte Carlo methods to draw samples from the posterior. We demonstrate this method on two core problems in computer vision—structure from motion and colour constancy. These examples illustrate a samplers producing useful representations for very large problems. We demonstrate that the sampled representations are trustworthy, using consistency checks in the experimental design. The sampling solution to structure from motion is strictly better than the factorisation approach, because: it reports uncertainty on structure and position measurements in a direct way; it can identify tracking errors; and its estimates of covariance in marginal point position are reliable. Our colour constancy solution is strictly better than competing approaches, because: it reports uncertainty on surface colour and illuminant measurements in a direct way; it incorporates all available constraints on surface reflectance and on illumination in a direct way; and it integrates a spatial model of reflectance and illumination distribution with a rendering model in a natural way. One advantage of a sampled representation is that it can be resampled to take into account other information. We demonstrate the effect of knowing that, in our colour constancy example, a surface viewed in two different images is in fact the same object. We conclude with a general discussion of the strengths and weaknesses of the sampling paradigm as a tool for computer vision.

Keywords: Markov chain Monte Carlo, colour constancy, structure from motion

1. Introduction

The Bayesian philosophy is that all information about a model is captured by a *posterior* distribution obtained using Bayes' rule:

$$\begin{aligned}\text{posterior} &= P(\text{world} \mid \text{observations})\\ &\propto P(\text{observations} \mid \text{world})\, P(\text{world})\end{aligned}$$

where the prior $P(\text{world})$ is the probability density of the state of the world in the absence of observations. Many examples suggest that, *when computational difficulties can be sidestepped,* the Bayesian philosophy leads to excellent and effective use of data (e.g. expositions in Carlin and Louis (1996), Gelman et al. (1995), Grenander (1983), and Grenander (1993); examples of the use of Bayesian inference in the vision literature include (Binford and Levitt, 1994; Chou and Brown, 1990; Huang et al., 1994; Jolly et al., 1996; Maybank and Sturm, 1999; Noble and Mundy, 1993; Pavlovic et al., 1999a, 1999b; Sarkar and Boyer, 1992, 1994; Sullivan et al., 1999; Yuille and Coughlan, 1999; Zhu et al., 2000, among others).

A probability distribution is, in essence, a device for computing expectations. The problems we are interested in typically involve an important continuous component, meaning that computing expectations involves estimating integrals, usually over high dimensional domains. One useful technique is to represent the posterior by drawing a large number of samples from that distribution. These samples can then be used to estimate any expectation with respect to that posterior.

For example, if we wished to decide whether to fight with or flee from an attacker, we would draw samples from the posterior on the outcome and estimate expected utilities for each decision (as averages of the utilities over the samples) and then choose the decision with the best utility.

Sampling algorithms are more general than random search for MAP interpretations precisely because the results give an approximate representation of the entire posterior. This means that, for example, we can estimate the covariance of the posterior; resample the samples to incorporate new information; engage in multiple calculations for different decisions using distinct utilities, etc. Sampling is *in principle* simple and general, *if* samples can be drawn from the posterior efficiently.

This paper demonstrates the strengths and weaknesses of sampling methods using two core vision problems as examples: structure from motion (Section 2); and colour constancy (Section 3).

Notation: we write ϵ for a vector, whose i'th component is ϵ_i and $\mathcal{M}$ for a matrix whose i, j'th component is M_{ij}. Sampler jargon that may be unfamiliar is shown in italics when first introduced.

1.1. Simple Sampling Algorithms

For some probability distributions, direct algorithms exist for drawing samples (e.g. Ripley, 1987); we are seldom lucky enough to have a posterior of this type. *Rejection sampling* is appropriate for some distributions. Assume that we wish to draw samples from $p(x)$ and have a proposal distribution $q(x)$, from which we can draw samples easily. Assume also we know some constant, k, such that $kq(x) \geq p(x)$ for all x; we can draw a sample from $p(x)$ by drawing a sample x_0 from $q(x)$, and then accepting the sample with probability $p(x_0)/(kq(x_0))$. In *importance sampling*, to draw a sample from $p(x)$, we first draw a large number of independent samples $\{s_1, \ldots, s_n\}$ from a proposal distribution $q(x)$, and then set $s = s_i$ with probability proportional to $w_i = \frac{p(x)}{q(x)}$. As $n \to \infty$, the distribution for the sample s will approach $p(x)$. Both rejection sampling and importance sampling methods can be wildly inefficient if $q(x)$ approximates $p(x)$ poorly (the usual case in high dimensions); in some such cases, a collection of different $q(x)$'s can be pasted together to obtain a better approximation (e.g. Gamerman, 1997).

1.2. Markov Chain Monte Carlo—the Metropolis-Hastings Algorithm

Markov chain Monte Carlo methods (Gamerman, 1997; Gilks et al., 1996c) are the standard methods for sampling complex distributions. In this method, one constructs a Markov chain whose stationary distribution is the target distribution. A new sample can be obtained from an old one, by advancing the Markov chain.

The Metropolis-Hastings algorithm is a technique for constructing a Markov chain that has a particular desired stationary distribution. Assume that we have a distribution π from which we would like to generate samples. We would like to build a Markov chain which has π as a stationary distribution.

The algorithm will produce a sequence of samples $X_1, \ldots, X_n$, by taking a sample X_i and proposing a revised version, X_i'. The next element of the sequence X_{i+1} will be X_i' with probability $\alpha(X_i, X_{i+1})$; otherwise, it will be X_i. We will give the form of α below.

The proposal process is random, too. In particular, there is a proposal distribution which gives the probability of proposing X_i' from X_i. This can be written $P(X_i \to X_i')$. Note that the proposal distribution is a function of X_i', and may be a function of X_i. Now we assume that if $\pi(u)$ is non-zero, then there are some values v such that $P(v \to u)$ is non-zero, too.

In this case, we have that

$$\alpha = \max\left(1, \frac{P(X_i' \to X_i)\pi(X_i')}{P(X_i \to X_i')\pi(X_i)}\right)$$

Notice that this expression is qualitatively sensible. If the chain is at a point where π has a very low value and at the new point π has a very high value *and* the forward and backward proposal probabilities are about equal, then the new point will be accepted with high probability. If the chain is at a point where π has a very high value and the proposal process has a high probability of suggesting points with a very low value of π, it is likely to stay at that point. Finally, if a point which has high value of π is proposed disproportionately often, it is less likely to be accepted.

A good way to think about the Metropolis-Hastings algorithm is that it is an improved version of the "hypothesize and test" process that is common in vision. Metropolis-Hastings suggests various hypotheses which, depending on the result of a bookkeeping exercise, are accepted or rejected. This process yields the sequence $X_1, \ldots, X_n$. However, for Metropolis-Hastings the sequence of hypotheses has very significant semantics; assuming technical conditions on the proposal process (expounded in, for example, Gamerman, 1997; Gilks et al., 1996a; Roberts, 1996; Tierney, 1996; these conditions are usually fairly easily met and all our samplers meet them), once sufficient iterations have completed, all subsequent X_i are samples drawn from $\pi(X)$.

1.3. Burn-in and Mixing

Generally, an MCMC method needs to produce some number of samples to "forget" its start point. The number of iterations required to achieve this is often called the *burn-in* time. The burn-in may be extremely long for a poorly designed sampler. There are a very small number of samplers known to have a short burn-in time (e.g. Jerrum and Sinclair, 1996).

Once a sampler has burnt in, the sequence of samples it produces may or may not be correlated; if this correlation is low, the method is said to *mix* well. It is desirable to have an algorithm that burns in quickly, and mixes well. Burn-in and mixing are related to the dynamics of the underlying Markov chain. One way to show that a sampler mixes quickly is to prove that, for any decomposition of its domain into two disjoint sets A and B, the conditional probability that the sampler goes to B given it is in set A, is high. Such proofs of fast mixing exist for a small number of cases, but require substantial art (e.g. Jerrum and Sinclair, 1996). We are aware of no proof that a sampler used for vision problems is fast mixing. In our examples, as in the vast majority of cases, the algorithm is used without such proofs; we show a variety of consistency checks that suggest that the algorithm has converged.

1.4. The Attractions of MCMC

It is known how to apply this algorithm when the domain of support is complicated (for example, samples may be drawn when the domain of support of the posterior consists of several different spaces of different dimensions, Green, 1995; Richardson and Green, 1997). There are numerous variants to the basic algorithm, some of which combine deterministic dynamics with random search in the hope of better mixing (e.g. see the review in Neal (1993)).

The advantage of viewing Metropolis-Hastings algorithms as a souped up hypothesize and test process is that it suggests how to build proposal mechanisms. A natural strategy is to take current vision algorithms and make them produce probabilistic outputs. This approach is illustrated in Section 3.2; Zhu et al. (2000) have used it successfully for some recognition problems. A really attractive feature is that we can use different, possibly incompatible algorithms as distinct sources of proposals, and the samples we obtain represent the posterior incorporating *all* available measurements.

Quite often in practice it is easy to come up with a function f proportional to the posterior. In this case, the posterior is

$$\frac{f}{\int_D f(u)\, du}$$

but the integral—the normalizing constant—can be very difficult to compute (the best way to do it is to use a sampling method). An attractive feature of the Metropolis Hastings algorithm is that we need not know the normalizing constant for the distribution (because the constant is cancelled by the ratio).

1.5. Techniques for Building Practical MCMC Samplers

It is easy to build a sampler using the Metropolis-Hastings algorithm. It seems to be very hard to build a *good* sampler—one that burns in quickly, mixes well, and gives a trustworthy picture of the posterior—using that algorithm. We describe a variety of techniques for building samplers, and conclude with a discussion of possible sanity checks.

1.5.1. Gibbs Samplers. It is quite common to encounter situations where the target distribution has a non standard form, but is standard when groups of variables have fixed values (this occurs in vision problems; see Sections, 2.3 and 3.2). In this case, it is natural to adopt a proposal mechanism that fixes one set of variables and draws a sample from the full conditional distribution on the other set, and vice versa. This very useful technique is known as *Gibbs sampling* (named by Geman and Geman (1984) but apparently due to the statistical physics literature, where it was known as the *heat bath algorithm*, Gilks et al., 1996b, p. 12). Usually, the group of variables to be sampled is chosen at random, and sufficient samples are drawn so that each group of variables is visited many times.

Gibbs sampling is very easy to implement. There is one considerable danger, which is often quite difficult to avoid. If the groups of variables are strongly

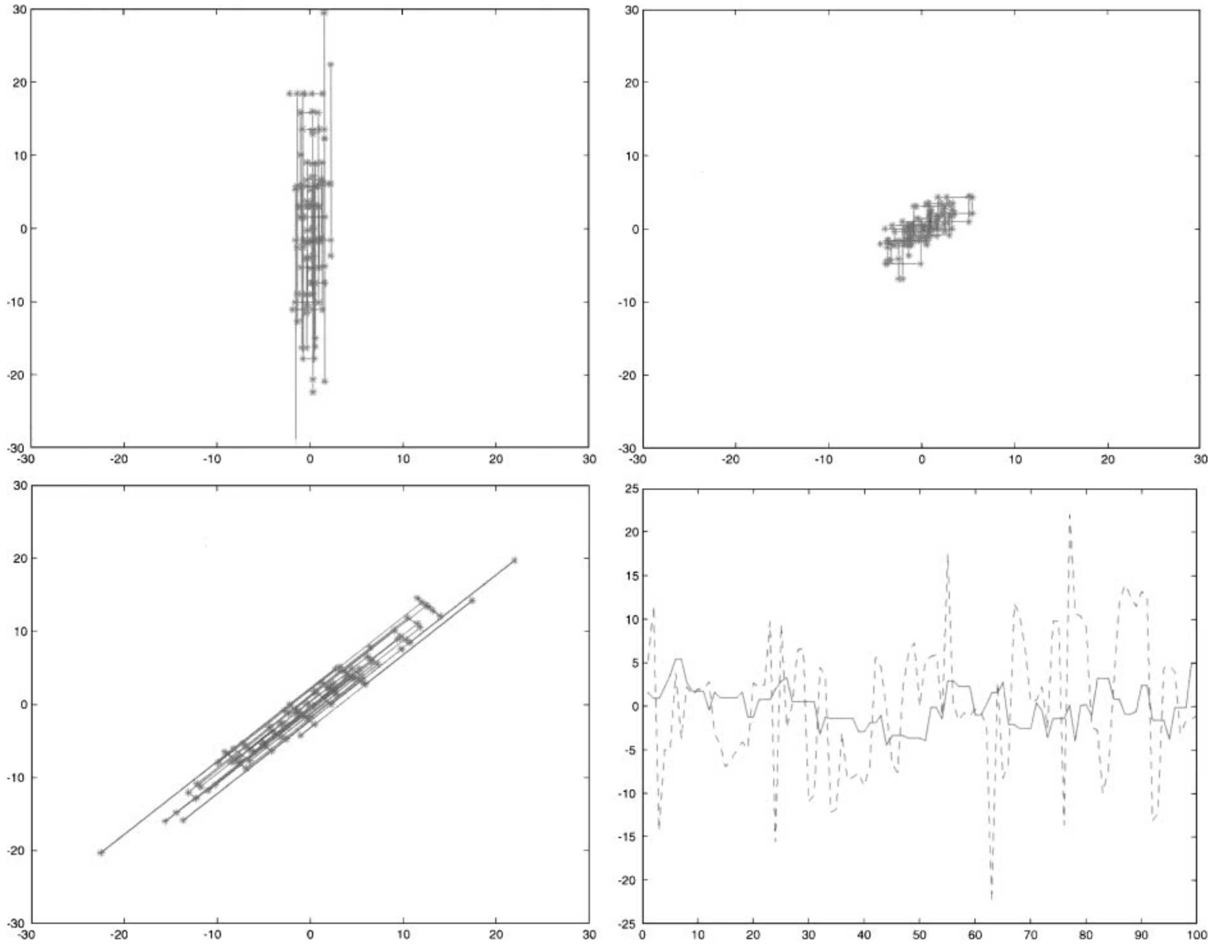


Figure 1. Correlated variables cause Gibbs samplers to behave badly. The figure on the *top left* shows 100 samples drawn from a Gibbs sampler for two independent normal random variables, one with variance one and the other with variance ten. The stars indicate the samples; the line segments indicate the order in which the samples were drawn. Note that the sampler makes quite large vertical moves (because the variance in this direction is large). The figure on the *top right* shows 100 samples drawn from this distribution, now rotated by 45°, using a Gibbs sampler. In this case, the sampler can make only relatively small vertical and horizontal moves, and so the position of the samples changes relatively slowly; the 100 samples in the graph on the *bottom left*, which consist of those of the first graph rotated by 45°, give a much better picture of the distribution. On the *bottom right*, the x-coordinate for the samples drawn from the second sampler (solid line) and the x-coordinates of the third figure (dashed line). The solid curve (correctly) suggests that the samples drawn from the second sampler are quite strongly correlated.

correlated, then a Gibbs sampler can mix very badly indeed. The effect is well known (for a full discussion, see for example, Gilks and Roberts, 1996) and easily illustrated (see Fig. 1).

1.5.2. The Hybrid Monte Carlo Method. A common difficulty with sampling methods is that the state of the sampler appears to perform a slightly biased random walk. The difficulty with random walk is that it takes a long time to move any distance along a domain, meaning that if the sampler is started at a point a long way from the mode of the distribution, it will take a long time before it reaches the mode. From our perspective, it is extremely important to have a representation of the distribution around the mode.

Hybrid Monte Carlo is a method for making proposals that causes the state of the sampler to move rather quickly to the mode, and then explore it. The method is due to Duane et al. (1987) (and described in detail in Neal (1993)). Write the state of the sampler as $\mathbf{q}$. The method requires that the target distribution can be written as

$$\pi(\mathbf{q}) = \exp\{-U(\mathbf{q})\}$$

Now let us think of U as a potential function; the state of the sampler will be the state of a particle of mass m subject to this potential function. This state can be determined by considering the momentum of the particle $\mathbf{p}$ and writing a Hamiltonian for the particle:

$$H(\mathbf{q}, \mathbf{p}) = U(\mathbf{q}) + \frac{\mathbf{p}^T\mathbf{p}}{2m}$$

We now need to integrate Hamilton's equations

$$\begin{aligned} \frac{\partial \mathbf{q}}{\partial t} &= \frac{1}{m}\mathbf{p} \\ \frac{\partial \mathbf{p}}{\partial t} &= -\nabla_{\mathbf{q}} U \end{aligned}$$

to determine the state of the particle. This temporary excursion into mechanics is actually justified, because we can exponentiate the negative Hamiltonian of the particle to get

$$\begin{aligned} \pi'(\mathbf{q}, \mathbf{p}) &= \exp\{-H(\mathbf{q}, \mathbf{p})\} \\ &= \pi(\mathbf{q}) \exp\left\{-\frac{\mathbf{p}^T\mathbf{p}}{2m}\right\} \end{aligned}$$

which is a new target distribution for a larger set of random variables. We now have two proposal moves:

1. Advance time in our particle model by some randomly chosen amount, either forwards or backwards. This updates both $\mathbf{q}$ and $\mathbf{p}$. As long as we use a symplectic integrator, the extent of the advance is uniform and random, and the choice of forward or backward is random, the accept probability is one.
2. Fix $\mathbf{q}$ and draw a sample for $\mathbf{p}$ from the full conditional. This is easy, because the full conditional distribution in $\mathbf{p}$ is normal and is independent of $\mathbf{q}$.

This sampler has very attractive *qualitative* behaviour. If the state is at a relatively large value of U, then the first type of move will travel quickly down the gradient of U to smaller values, while building up momentum. But the second move then discards this momentum; so we have a sampler that should move quickly to a mode—where U is small—and then move around exploring the mode under the influence of the random choice of momenta. Good values of the particle's mass and of the range of time values must be chosen by experiment.

In practice, the hybrid method seems to be useful for continuous problems. It is very easy to implement for the colour constancy example given above, and has been successfully used on a variety of other continuous problems (Neal, 1993).

1.6. MCMC and Random Search in Vision

Markov chain Monte Carlo has appeared in the vision literature in various forms. One common use is to attempt to obtain an MAP estimate by random search, usually using the Metropolis-Hastings algorithm (e.g. Geman and Geman, 1984; Geman and Graffigne, 1986). The Markov random field model is a spatial model which gives a posterior on image labellings given measurements as a function of measurement values and local patterns of pixel labels (so-called "clique potentials"; the topic is reviewed in Li (1995)). A standard method for estimating MAP labellings is to use an annealed version of the Metropolis-Hastings algorithm, where the posterior to be sampled is a function of a parameter that changes during the sampling process. This parameter is often thought of as temperature; the intent is that for high values of the parameter, the posterior has only one mode, and as the temperature is reduced the state of the sampler will get stuck in that mode, thereby obtaining a global extremum. It is not possible to guarantee in practice that this occurs, and the algorithm has a rather mixed reputation (Collins et al., 1988; Golden and Skiscim, 1986).

The notion of using a sampling method to perform inference on a generative model of an image pattern appears to be due to Grenander (1983). Few successful examples appear in the literature. In Jolly et al. (1996), an annealing method is used to estimate an MAP solution for the configuration and motion of a motor car template in an image. In Zhu (1998), a random search method is used to find a medial axis transform. In Zhu et al. (2000), an MCMC method is used to find simple shapes and road signs. In Green (1996), MCMC is used to perform inference in various vision-like situations, including reconstruction from single photon emission computed tomography data and finding a polygonal template of a duck in heavy spatial noise. In Phillips and Smith (1996), inference is performed on a hierarchical model to find faces, and a version of MCMC is used to find an unknown number of disks. Templates are used for restoration in Amit et al. (1991). Gibbs samplers are quite widely used for reconstruction (Geman and Geman, 1984; Geman and Graffigne, 1986; Zhu et al., 1998).

Random search is now a standard method for estimating the fundamental matrix in structure from motion problems; a review appears in Torr and Murray (1997). RANSAC—an algorithm for robust fitting, due to Fischler and Bolles (1981) and appearing in the statistical literature as Rousseeuw (1987)—proposes small sets of correspondences uniformly at random, fits a fundamental matrix to each set, and accepts the set whose fit gives the largest number of correspondences with a sufficiently small residual. The number of sets is chosen to ensure some high probability that a correct set is found. The main advantage of an MCMC method over RANSAC is that an MCMC method can produce a series of hypotheses with meaningful semantics—indicating, for example, the posterior probability that a particular point is an outlier, or the posterior probability that a pair of measurements come from a single point.

1.6.1. Particle Filtering (or Condensation, or "Survival of the Fittest") and Resampling. The most substantial impact of sampling algorithms in vision has been the use of resampling algorithms in tracking. The best known algorithm is known as *condensation* in the vision community (Blake and Isard, 1998), *survival of the fittest* in the AI community (Kanazawa et al., 1995), and *particle filtering* in the statistical signal processing community, where it originated (Carpenter et al., 1999; Kitagawa, 1987). A wide range of variants and of applications of particle filtering are described in a forthcoming book (Doucet et al., 2001). This algorithm is a modification of factored sampling: one draws samples from a prior (which represents the state of the world up to the $k-1$'th measurement), propagates these samples through a dynamical model, and then weights them using the posterior incorporating the k'th measurement. This set of weighted samples provides a representation of the prior for the next iteration. The algorithm is fast and efficient, and is now quite widely applied for low-dimensional problems.

The attraction of resampling algorithms is that they can be used to incorporate "new information." In tracking applications, new information comes because a new frame, with new measurements, has arrived. New information may come from other sources. In the colour constancy example, we assume that the algorithm is told that two patches in two different images are the same colour (this might occur because a recognition algorithm has a good match to the geometry, and knows the patches represent the same object). This information strongly constrains the inferred colours for other patches in each view (Section 3).

In recognition applications one often encounters some form of hierarchical model, which again suggests resampling. In Ioffe and Forsyth (1999), a sampler is used to label groups of image segments, using their consistency with observed human kinematics. The human model used has nine segments. It is foolish to attempt to label all nine segment groups; instead, their algorithm uses a sampler to label individual segments with a frequency proportional to the posterior probability of that label given the image data. The set of individual segment labels is resampled to propose pairs of labels for pairs of segments, and so on. In this case, the new information is the use of an enhanced prior; the prior for pairs of labels emphasizes pairs of segments that lie in particular configurations, a property that is meaningless for single segments.

2. Example: Large Scale Sampling for Bayesian Structure from Motion

Structure from motion is the problem of inferring some description of geometry from a sequence of images. The problem has a long history and a huge literature; space does not allow a comprehensive review, but see Beardsley et al. (1997), Faugeras et al. (1998), Faugeras and Robert (1996), Gool and Zisserman (1997), and Hartley and Zisserman (2000). Accurate solutions to structure from motion are attractive, because the technique can be used to generate models for rendering virtual environments (e.g. Debevec et al., 1996; Faugeras et al., 1998; Gool and Zisserman, 1997; Tomasi and Kanade, 1992).

2.1. Structure from Motion by Matrix Factorisation

Assume m distinct views of n points are given; correspondences are known. In the influential Tomasi-Kanade formulation of structure from motion (Tomasi and Kanade, 1992), these data are arranged into a $2m \times n$ matrix of measurements $\mathcal{D}$ which must factor as $\mathcal{D} = \mathcal{U}\mathcal{V}$, where $\mathcal{U}$ represents the camera positions and $\mathcal{V}$ represents point positions. An affine transform $\mathcal{A}$ is determined such that $\mathcal{U}\mathcal{A}$ minimises a set of constraints associated with a camera, and $\mathcal{A}^{-1}\mathcal{V}$ then represents Euclidean structure.

In practice, factorisation is achieved using a singular value decomposition. This is a maximum likelihood method if an isotropic Gaussian error model is adopted; for an anisotropic Gaussian error model, see Morris and Kanade (1998). The formalism has been applied to various camera models (Poelman, 1993;

Tomasi and Kanade, 1992; Triggs, 1995); missing data points can be interpolated from known points (Jacobs, 1997; Tomasi and Kanade, 1992); methods for motion segmentation exist (Costeira and Kanade, 1998); and methods for lines and similar primitives are known (Morris and Kanade, 1998). There are noise estimates for recovered structure (Morris and Kanade, 1998). These assume that errors in the estimates of structure are independent, an assumption that the authors acknowledge is not always sustainable.

The factorisation method has one important weakness. Because the algorithm has two separate stages, it does not allow any payoff between model error—the extent to which the recovered model violates the required set of camera constraints—and measurement error—the extent to which model predictions correspond to data observations. This means that the model cannot be used to identify measurement problems (for example, tracker errors as in Fig. 5), and so is subject to reconstruction errors caused by incorporating erroneous measurements. This is a property of the algorithm, rather than of the problem; because $\mathcal{U}$ and $\mathcal{V}$ have relatively few degrees of freedom compared with $\mathcal{D}$, it should be possible to identify and ignore many unreliable measurements if the full force of the model is employed. Recent work by Dellaert et al. has shown how strongly the model constrains the data; they use a sampling method to average over all correspondences, weighting them by consistency with measured data, and obtaining a satisfactory reconstruction. Their method removes the need to compute correspondences from structure from motion problems (Dellaert et al., 2000).

2.2. The Posterior on Structure and Motion

It is useful to think of Bayesian models as generative models (e.g. Grenander, 1983). In a generative structure from motion model, $\mathcal{U}$ and $\mathcal{V}$ are drawn from appropriate priors. Then $\mathcal{D}$ is obtained by adding noise to $\mathcal{U}\mathcal{V}$. We assume that noise is obtained from a mixture model; with some large probability, Gaussian noise is used, and with a small probability, the measurement value is replaced with a uniform random variable.

The priors on $\mathcal{U}$ and $\mathcal{V}$ are obtained from constraints on camera structure. We do not fix the origin of the coordinate system, and represent points in homogenous coordinates, so our $\mathcal{U}$ and $\mathcal{V}$ have dimensions $2m \times 4$ and $4 \times n$ respectively. We assume a scaled orthographic viewing model with unknown scale that varies from frame to frame.

All this yields a vector of constraint equations

$$\mathbf{C}(\mathcal{U}, \mathcal{V}) = \mathbf{0}$$

which contains elements of the form

$$\sum_{j=1}^{3}(u_{i,j})^2 - \sum_{j=1}^{3}(u_{i+m,j})^2$$

(expressing the fact that the camera basis consists of elements of the same length),

$$\sum_{j=1}^{3}(u_{i,j}u_{i+m,j})$$

(expressing the fact that the camera basis elements are perpendicular), and

$$v_{j,4} - 1$$

(from the homogenous coordinates). A natural prior to use is proportional to

$$\exp\left(\frac{-\mathbf{C}^T(\mathcal{U}, \mathcal{V})\mathbf{C}(\mathcal{U}, \mathcal{V})}{2\sigma^2_{\text{constraint}}}\right)$$

This prior penalises violations of the constraints quite strongly, but allows constraint violations to be paid off one against the other. This approach is in essence a penalty method. An alternative is to insist that the prior is uniform if the constraints are all satisfied and zero otherwise—in practice, this would involve constructing a parametrisation for the domain on which the prior is non-zero, and working with that parametrisation. This approach is numerically more complex to implement; it also has the disadvantage that one is imposing constraints that may, in fact, be violated (i.e. the scaled orthography model may not be sufficient; the imaging element may be misaligned with respect to the lens, so that the camera basis consists of elements of slightly different length, etc.).

We can now write a posterior model. Recall that the noise process is a mixture of two processes: the first adds Gaussian noise, and the second replaces the measurement value with a uniform random variable. We introduce a set of discrete mask bits, one per measurement, in a matrix $\mathcal{M}$; these mask bits determine by which noise model a measurement is affected. A mask bit will be 1 for a “good” measurement (i.e. one affected by isotropic Gaussian noise), and 0 for a “bad” measurement (i.e. one which contains no information about the model). These bits should be compared with the mask bits used in fitting mixture models using EM (see the discussion in McLachlan and Krishnan (1996), and with the boundary processes used in, among others, Blake and Zisserman, 1987; Mumford and Shah, 1989). We introduce a prior on $\mathcal{M}, \pi(\mathcal{M})$, which is zero for matrices that have fewer than k non-zero elements in some row or column, and uniform otherwise; this prior ensures that we do not attempt inference for situations where we have insufficient measurements.

The likelihood is then $P(\mathcal{D}|\mathcal{U}, \mathcal{V}, \mathcal{M})$, which is proportional to the exponential of

$$-\left\{ \sum_{i,j} \frac{(d_{ij} - \sum_k u_{ik} v_{kj})^2 m_{ij}}{2\sigma^2_{\text{meas}}} + \frac{(1 - m_{ij})}{2\sigma^2_{\text{bad}}} \right\}$$

and the posterior is proportional to:

$$P(\mathcal{D} \mid \mathcal{U}, \mathcal{V}, \mathcal{M}) \times \exp\left(\frac{-\mathbf{C}^T(\mathcal{U}, \mathcal{V})\mathbf{C}(\mathcal{U}, \mathcal{V})}{2\sigma^2_{\text{constraint}}} \right) \pi(\mathcal{M})$$

Notice that the maximum of the posterior could well not occur at the maximum of the likelihood, because although the factorisation might fit the data well, the $\mathcal{U}$ factor may satisfy the camera constraints poorly.

2.3. Sampling the Structure from Motion Model

This formulation contains both a discrete and a continuous component. It is natural to consider using a Gibbs sampler, sampling from the full conditional on point positions given fixed camera positions, and from the full conditional on camera positions given fixed point positions. This works poorly, because the variables are very highly correlated—a tiny shift in a point position given fixed camera positions tends to result in a large error. Instead, the continuous variables are sampled using the hybrid method described in Section 1.2; discrete variables are sampled from the full conditional using a strategy that proposes inverting 5% of the bits, randomly chosen, at a time. Hybrid MCMC moves are proposed with probability 0.7 and discrete variable moves are proposed with probability 0.3.

3. Example: Sampling an Unknown Number of Components for Bayesian Colour Constancy

The image appearance of a set of surfaces is affected both by the reflectance of the surfaces and by the spectral radiance of the illuminating light. Recovering a representation of the surface reflectance from image information is called *colour constancy.* Computational models customarily model surface reflectances and illuminant spectra by a finite weighted sum of basis functions and use a variety of cues to recover reflectance, including (but not limited to!): specular reflections (Lee, 1986); constant average reflectance (Buchsbaum, 1980); illuminant spatial frequency (Land and McCann, 1971); low-dimensional families of surfaces (Maloney and Wandell, 1986) and physical constraints on reflectance and illumination coefficients (Forsyth, 1990; Finlayson, 1996). Each cue has well-known strengths and weaknesses. The most complete recent study appears to be Brainard and Freeman (1997), which uses the cues to make Bayesian decisions that maximise expected utility, and compares the quality of the decision; inaccurate decisions confound recognition (Funt et al., 1998).

3.1. The Probabilistic Model

We assume that surfaces are flat, so that there is no shading variation due to surface orientation and no interreflection. There are four components to our model:

- **A viewing model:** we assume a perspective view of a flat, frontal surface, with the focal point positioned above the center of the surface. As spatial resolution is not a major issue here, we work on a 50 × 50 pixel grid for speed.
- **A spatial model of surface reflectances:** because spatial statistics is not the primary focus of this paper, we use a model where reflectances are constant in a grid of boxes, where the grid edges are not known in advance. A natural improvement would be the random polygon tesselation of Green (1996).
- **A spatial model of illumination:** for the work described in this paper, we assume that there is a single point source whose position is uniformly distributed within a volume around the viewed surface.
- **A rendering model:** which determines the receptor responses resulting from a particular choice of illuminant and surface reflectance; this follows from standard considerations.

3.1.1. The Rendering Model. We model surface reflectances as a sum of basis functions $\phi_j(\lambda)$, and assume that reflectances are piecewise constant:

$$s(x, y, \lambda) = \sum_{j=0}^{n_s} \sigma_j(x, y)\phi_j(\lambda)$$

Here $\sigma_j(x, y)$ are a set of coefficients that vary over space according to the spatial model.

Similarly, we model illuminants as a sum of basis functions ψ_i and assume that the spatial variation is given by the presence of a single point source positioned at $\mathbf{p}$. The diffuse component due to the source

$$e_d(x, p, \lambda, \mathbf{p}) = d(x, y, \mathbf{p}) \sum_{i=0}^{n_e} \epsilon_i \psi_i(\lambda)$$

where ϵ_i are the coefficients of each basis function and $d(x, y, \mathbf{p})$ is a gain term that represents the change in brightness of the source over the area viewed. The specular component due to the source is:

$$e_m(x, p, \lambda, \mathbf{p}) = m(x, y, \mathbf{p}) \sum_{i=0}^{n_e} \epsilon_i \psi_i(\lambda)$$

where $m(x, y, \mathbf{p})$ is a gain term that represents the change in specular component over the area viewed.

Standard considerations yield a model of the k'th receptor response as:

$$\begin{aligned} p_k(x, y) = d(x, y, \mathbf{p}) \sum_{i,j} g_{ijk} \epsilon_i \sigma_j(x, y) \\ + m(x, y, \mathbf{p}) \sum_i h_{ik} \epsilon_i \end{aligned}$$

where

$$g_{ijk} = \int \rho k(\lambda) \psi_i(\lambda) \phi_i(\lambda) \, d\lambda$$

and

$$h_{ik} = \int \rho k(\lambda) \psi_i(\lambda) \, d\lambda$$

and $\rho_k(\lambda)$is the sensitivity of the k'th receptor class. The illuminant terms $d(x, y, \mathbf{p})$ and $m(x, y, \mathbf{p})$ follow from the point source model; $m(x, y, \mathbf{p})$ is obtained using Phong's model of specularities.

We write any prior probability distribution as π. Our model of the process by which an image is generated is then:

- sample the number of reflectance steps in x and in y (k_x and k_y respectively) from the prior $\pi(k_x, k_y) = \pi(k_x)\pi(k_y)$.
- now sample the position of the steps ($\mathbf{e}_x$ and $\mathbf{e}_y$ respectively) from the prior $\pi(\mathbf{e}_x, \mathbf{e}_y \mid k_x, k_y) = \pi(\mathbf{e}_x \mid k_x)\pi(\mathbf{e}_y \mid k_y)$;
- for each tile, sample the reflectance ($\sigma^{(m)}$ for the m'th tile) for that tile from the prior $\pi(\sigma^{(m)})$;
- sample the illuminant coefficients ϵ from the prior $\pi(\epsilon)$;
- sample the illuminant position $\mathbf{p}$ from the prior $\pi(\mathbf{p})$;
- and rendser the image, adding Gaussian noise of known standard deviation σ_{cc} to the value of each pixel.

This gives a likelihood,

$$P\big(\text{image} \mid k_x, k_y, \mathbf{e}_x, \mathbf{e}_y, \sigma^{(1)}, \ldots, \sigma^{(k_x k_y)}, \epsilon, \mathbf{p}\big)$$

The posterior is proportional to:

$$\begin{aligned} P\ \big(\text{image} \mid k_x, k_y, \mathbf{e}_x, \mathbf{e}_y, \sigma^{(1)}, \ldots, \sigma^{(k_x k_y)}, \epsilon_i, \mathbf{p}\big) \\ \times \pi(\mathbf{e}_x \mid k_x)\pi(\mathbf{e}_y \mid k_y)\pi(k_x)\pi(k_y) \\ \times \pi(\epsilon_i)\pi(\mathbf{p}) \prod_{m \,\in\, \text{tiles}} \pi(\sigma^{(m)}) \end{aligned}$$

3.1.2. Priors and Practicalities. **The spatial model:** We specify the spatial model by giving the number of edges in the x and y direction separately, the position of the edges, and the reflectances within each block. We assume that there are no more than seven edges (8 patches) within each direction, purely for efficiency. The prior used is a Poisson distribution, censored to ensure that all values greater than seven have zero prior, and rescaled. Edge positions are chosen using a hard-core model: the first edge position is chosen uniformly; the second is chosen uniformly, so that the number of pixels between it and the first is never fewer than five; the third is chosen uniformly so that the number of pixels between it and the second and between it and the first is never fewer than five; and so on. This hard-core model ensures that edge are not so close together that pixel evidence between edges is moot.

Priors for reflectance and illumination: Surface reflectance functions can never be less than zero, nor greater than one. This means that the coefficients of these functions lie in a compact convex set. It is easy to obtain a representative subset of the family of planes that bounds this set, by sampling the basis functions at some set of wavelengths. Similarly, illuminant functions can never be less than zero, meaning that the coefficients of these functions lie in a convex cone. Again, this cone is easily approximated. These constraints on reflectance and illuminant coefficients are encoded in the prior. We use a prior that is constant within the constraint set and falls off exponentially with an estimate of distance from the constraint set. Because the constraint sets are convex, they can be expressed as a set of linear inequalities; for surface reflectance we have

$\mathcal{C}_s\sigma + \mathbf{b} > 0$ and for illuminant we have $\mathcal{C}_i\epsilon > \mathbf{0}$. If the coefficients in these inequalities are normalised (i.e. the rows of the matrices are unit vectors), then the largest negative value of these inequalities is an estimate of distance to the constraint set.

We use six basis elements for illumination and reflectance so that we can have (for example) surfaces that look different under one light and the same under another light. This phenomenon, known as *metamerism*, occurs in the real world; our exploration of ambiguities should represent the possibility. We *represent* surface colour by the colour of a surface rendered under a known, white light.

3.2. Sampling the Colour Constancy Model

Proposals are made by a mixture of five distinct moves, chosen at random. The probability of proposing a particular type of move is uniform, with the exception that when there are no edges, no deaths are proposed, and when the number of edges in a particular direction is at a maximum, no births are proposed. An important advantage to this approach is that, *within each move*, we can assume that the values of variables that we are not changing are correct, and so apply standard algorithms to estimate other values. Calculations are straightforward, along the lines of Green (1995).

Moving the light: Proposals for a new x, y position for the light are obtained by filtering the image. We apply a filter whose kernel has the same shape as a typical specularity and a zero mean to the r, g and b components separately; the responses are divided by mean intensity, and the sum of squared responses is rescaled to form a proposal distribution. The kernel itself is obtained by averaging a large number of specularities obtained using draws from the prior on illuminant position. Using image data to construct proposal distributions appears to lead to quite efficient samplers; it is also quite generally applicable, as Zhu et al. (2000) (who call it "data driven MCMC") point out. Proposals for a move of the light in z are uniform, within a small range of the current position. The real dataset has no specularities, and these moves have been demonstrated only for synthetic data.

Birth of an edge: For each direction, we apply a derivative of Gaussian filter to the red, green and blue components of the image and then divide the response by a weighted average of the local intensity; the result is squared and summed along the direction of interest. This is normalised to 0.8, and 0.2 of a uniform distribution is added. This process produces a proposal distribution that has strong peaks at each edge, and at the specularity, but does not completely exclude any legal edge point (Fig. 2). Again, we are using image information to construct an appropriate proposal process. For a given state, this proposal distribution is zeroed for points close to existing edges (for consistency with the hard core model), and a proposed new edge position is chosen from the result. Once the position has been chosen, we must **choose new reflectances** for each of the new patches created by the birth of an edge. Generally, if we give the two new patches reflectances that are similar to that of the old patch, we expect that there will be only a small change in the posterior; this is advantageous, because it encourages exploration. Currently, we average the receptor responses within each new patch, and then use the (known) illuminant to estimate a reflectance that comes as close as possible to achieving this average value, while lying within the constraint set. We then add a Gaussian random variable to the estimated reflectance value; currently, we use a vector of independent Gaussian components each of standard deviation 0.5 (the choice will depend on the basis fitted).

Death of an edge: The edge whose death is proposed is chosen uniformly at random. The death of an edge causes pairs of surface patches to be fused; the new reflectance for this fused region is obtained using the same mechanism as for a birth (i.e. the receptor responses are averaged, the known illuminant is used to estimate good reflectances for each patch, and a vector of independent Gaussian components each of standard deviation 0.5 is added to the result).

Moving an edge: An edge to move is chosen uniformly at random. Within the region of available points (governed by the hard-core model—the edge cannot get too close to the edges on either side of it) a new position is proposed uniformly at random. This is somewhat inefficient, compared with the use of filter energies as a proposal distribution. We use this mechanism to avoid a problem posed by a hard-core model; it can be difficult for a sampler to move out of the state where two edges are placed close together and on either side of a real edge. Neither edge can be moved to the real edge—the other repels it—and a new edge cannot be proposed in the right side; furthermore, there may be little advantage in killing either of the two edges. Proposing uniform moves alleviates this problem by increasing the possibility that one of the two edges will move away, so that the other can move onto the right spot.

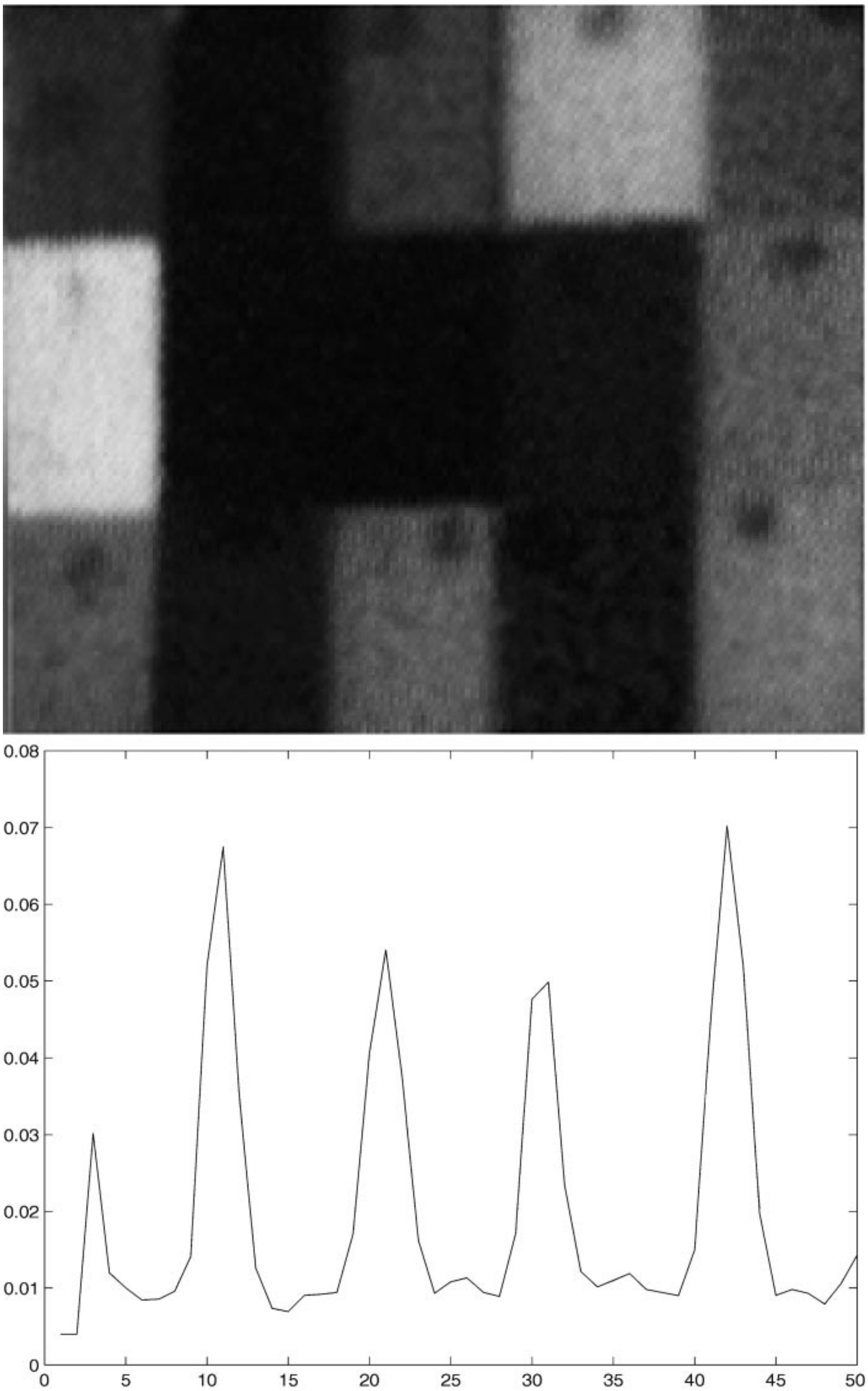


Figure 2. The proposal distribution for edge birth in the x direction for the Mondrian image shown. The proposal distribution is obtained by filtering the image, dividing the response by a weighted average of the local intensity, then summing down the y-direction. The result is normalised to 0.8, and 0.2 of a uniform distribution is added. Note that the filtering process leads to strong peaks near the edges; this means that the proposal process is relatively efficient, but does not completely rule out edges away from strong responses, if other evidence can be found for their presence (the likelihood component of the posterior).

Change reflectance and illumination: It is tempting to use a Gibbs sampler, but the chain moves extremely slowly if we do this. Instead, we sample reflectance and illumination simultaneously using the hybrid method of Section 1.2.

Poor behaviour by the Gibbs sampler can be explained as follows. Assume that the sampler has burnt in, which means that the current choice of surface reflectance and illuminant coefficients yields quite a good approximation to the original picture. Assume that we have fixed the surface reflectance coefficients and wish to change the illuminant coefficients. Now we expect that the normal distribution in illuminant coefficients has a mean somewhere close to the current value and a fairly narrow covariance, because any substantial change in the illuminant coefficients will lead to an image that is different from the original picture. This means that any change in the illuminant coefficients that results will be small. Similarly, if we fix the illuminant coefficients and sample the surface reflectance coefficients, we expect that the changes that result will be small.

4. Experimental Procedures

In each case, the sampler can be started at a state chosen at random, or at a state chosen by a start procedure (described in more detail in Section 5.4). The main difference between these methods is that choosing a start point tends to lead to a sampler that appears to burn in more quickly.

4.1. Structure from Motion

Results are obtained using the hotel dataset, courtesy of the Modeling by Videotaping group in the Robotics Institute, Carnegie Mellon University. We report two types of experiment: in the first, the sampler is run on that dataset; in the second, some small percentage of the points in this dataset are replaced with uniform random numbers in the range of the image coordinates. This represents large noise effects. Coordinates in this dataset appear to lie in the range 1–512. The algorithm appears to be quite well behaved for a rang of choices of constant. Values for the constants for Figs. 5, 6, 9 and 10 are $\sigma_{\text{meas}} = 1/\sqrt{2}$, $\sigma_{\text{constraint}} = 1/\sqrt{5000}$; σ_{bad} should be slightly larger than σ_{meas} (allowing points to range some distance from the measurement before the measurement has been disallowed) and we used $\sigma_{\text{meas}} = \sqrt{5} * \sigma_{\text{constraint}}$ for these figures. Experience suggests it is possible to use $\sigma_{\text{constraint}}$ very much smaller without apparently affecting the freedom with which the sampler mixes.

4.2. Colour Constancy

As Fig. 3 indicates, the sampler runs on synthetic images, and makes reasonable estimates of the position of the edges and the specularity and of illuminant and surface colours. In this case the basis and constraints are all known in advance. Applying the sampler to real data is more interesting. The data set shown in Fig. 8 consists of images originally used in Forsyth (1990).

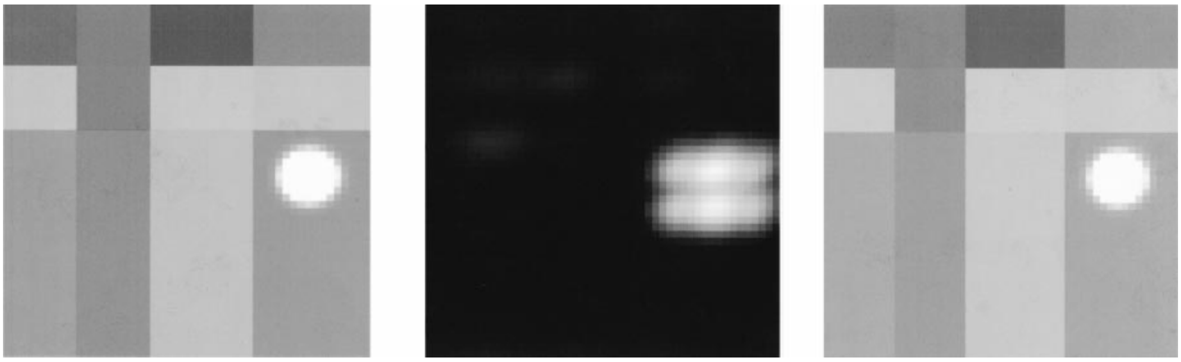


Figure 3. *Left:* a typical synthetic Mondrian, rendered using a linear intensity scale that thresholds the specularity. *Center:* the proposal distribution for x and y position of the specularity, obtained by image filtering and shown with the highest value white. *Right:* a rendering of a typical sample for this case, using the sample's illuminant; a successful sampler produces samples that look like the image. Results for real images are shown in colour in Fig. 8.

These are images of the same set of patches on a Mondrian of coloured paper patches, photographed under white, blue, yellow, purple, red and cyan light. There are no specularities, so we used a diffuse model for this data set.

The original data has been lost, so we used versions scanned from the paper; these images were displayed on a CRT, photographed from that display, subjected to four-colour printing and then scanned; it is remarkable that any constancy is possible under the circumstances. A basis was obtained using the bilinear fitting procedure of Marimont and Wandell (1992). Determining appropriate constraint regions is more difficult; we obtained a natural coordinate system using principal components, and then constructed a bounding box in this coordinate system. The box was grown 10% along each axis, on the understanding that none of the colours in the Mondrians of Forsyth (1990) were very deeply saturated. The red, green and blue receptor responses are represented by numbers in the range zero to one; we use $\sigma_{cc} = 1/64$, implying that only the top six bits in each receptor response are reliable.

5. Assessing the Experimental Results

Sections 2 and 3 phrased two standard vision problems as inference problems. These are quite nasty inference problems, with large numbers of both continuous and discrete variables. It is possible, as these sections indicated, to extract a representation of the posterior from these problems. Why do we believe that these representations are helpful? and how well do they compare with representations that other methods might offer?

Some cautions must be observed before making comparisons. Firstly, it is important to apply a reality check to the representations that the sampler produces, to determine if there is reason to believe that the sampler has burnt-in. Secondly, comparing a representation of a posterior given some data with the result of a method that reports a minimum error solution offers no more than a perfunctory error check. This is because the nature of the information produced by the two algorithms is different. The meaningful comparison is with other possible reports of the properties of the posterior. Here, no "gold standard" tests are available; there are no methods that are known to produce more accurate representations of a posterior density against which we can test a sampler. However, we *can* compare the representation produced by the sampler to methods that are significantly cheaper computationally.

5.1. Reality Checks: Has the Sampler Burnt in and is it Mixing?

There are convergence diagnostics for MCMC methods (e.g. see Besag et al., 1995; Roberts, 1992), but these can suggest convergence where none exists; it is easy to produce a chain that can pass these tests without having burnt in. Instead, we rely on general methods. Firstly, we check to ensure that the sampler can move to a near-maximal value of the posterior from any start position within a reasonable number of moves. Secondly, we check that the state of the sampler moves freely about the domain that is represented. Third, we have built various consistency checks into the experiments.

5.1.1. Structure from Motion. Figure 4 shows a series of samples drawn from the posterior for the structure from motion problem, with an indication of the order in which the samples were drawn, indicating that the sampler is mixing relatively well.

While the sampler's mixing rate does appear to be sufficient to give a reasonable estimate of structure of the posterior around its mode, it is clear that the sampler does not move around the whole domain freely. This posterior contains a discrete symmetry; for any fixed value of the mask bits, one can multiply $\mathcal{U}$ by a square root of the identity on the left and $\mathcal{V}$ by a square root of the identity on the right, and obtain the same value of the posterior. This creates no particular difficulty in practice, because these solutions are very widely isolated from one another. Our sampler does not move from peak to peak, because the probability that the hybrid method would obtain sufficient momentum to cross the very large regions of very low probability is effectively zero. This is in fact a desirable property; the symmetry means that accurate estimates of the mean value of $\mathcal{U}$ and $\mathcal{V}$ would be zero.

Consistency checks: In general, we expect that a sampler that is behaving properly should be able to identify correspondence errors and produce a stable representation. There are in fact a number of subtle tracker errors in the hotel sequence. Figure 5 shows that the sampler can identify these tracker errors. Figure 6 illustrates that large tracker errors, artificially inserted into the dataset for this purpose, can be identified, too.

5.1.2. Colour Constancy. The sampler described here has been run on many synthetic images where "ground truth" is known, and in each case reaches a small neighbourhood of ground truth from a randomly

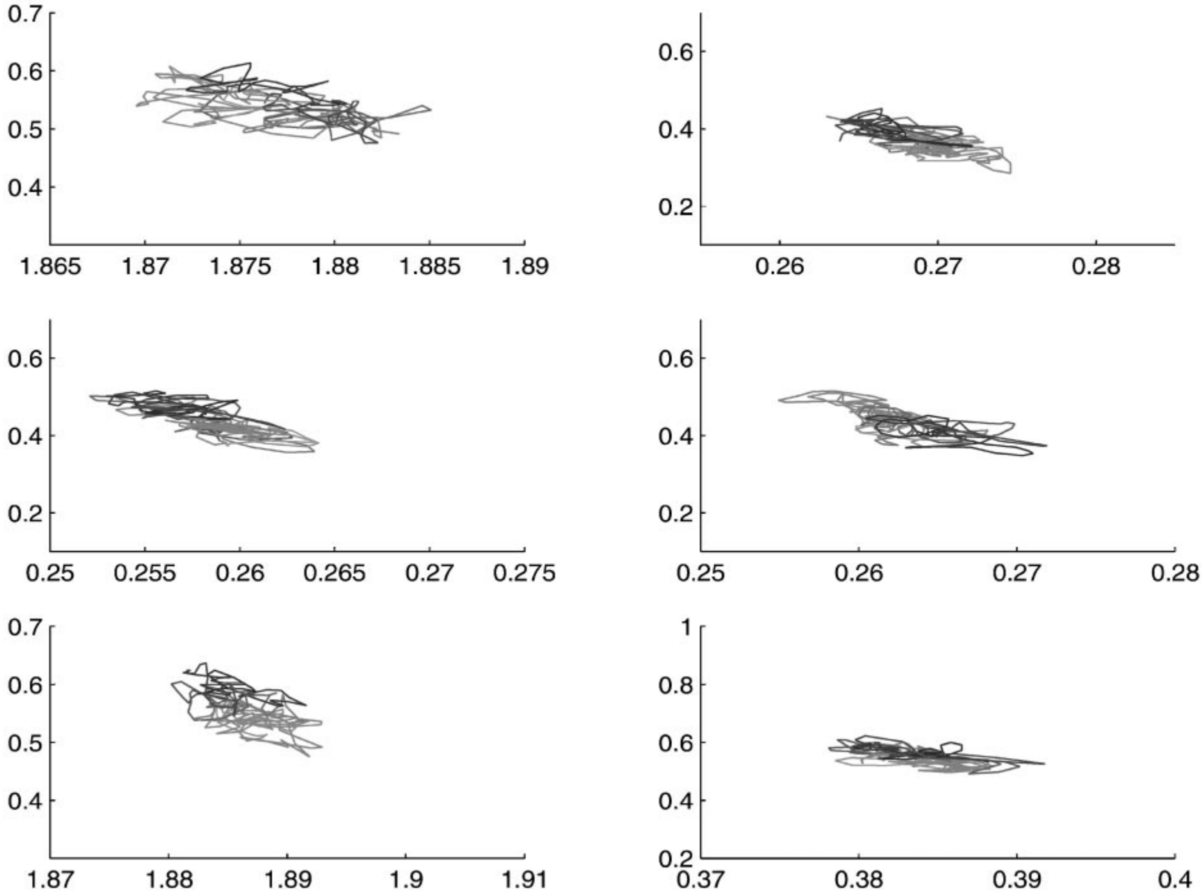


Figure 4. These plots illustrate the path taken through the state space by the structure from motion sampler. Each plot connects the position of a given point in every tenth sample, starting at the 100th. The paths have been coded with a grey level for clarity; the early samples are light, and the path moves through darker grey levels. The fact that these paths repeatedly cross themselves and return to the same regions suggests that the sampler is mixing rather freely.

selected start point—i.e. "burns in"—within about 1000 samples. The experimental data shown below suggests the sampler mixes well, because of the wide spread on the marginal densities on the reflectances.

Consistency checks: The sampler is run on six images of the same scene, but the fact that these images are of the same scene is not built into the model. The spread of samples for surface reflectance coefficients recovered for a particular surface in a particular image, is quite wide (see Fig. 8). However, if we compare the spread of samples for that surface for different images, the clusters overlap. This means that the representation is correctly encoding the fact that these surfaces could be very similar. In fact, as we shall see in Section 5.2, the representation encodes the fact that all surface patches could be very similar.

5.2. *Attractive Properties of Sampled Representations*

There are three attractive properties of the sampled representations we have derived:

- they provide a covariance estimate for inferred state;
- they can be resampled to incorporate new information;
- they appear to be stable to perturbations of the input data set.

We describe these properties below.

5.2.1. Covariance. The samplers described produce a representation of the posterior probability distribution,

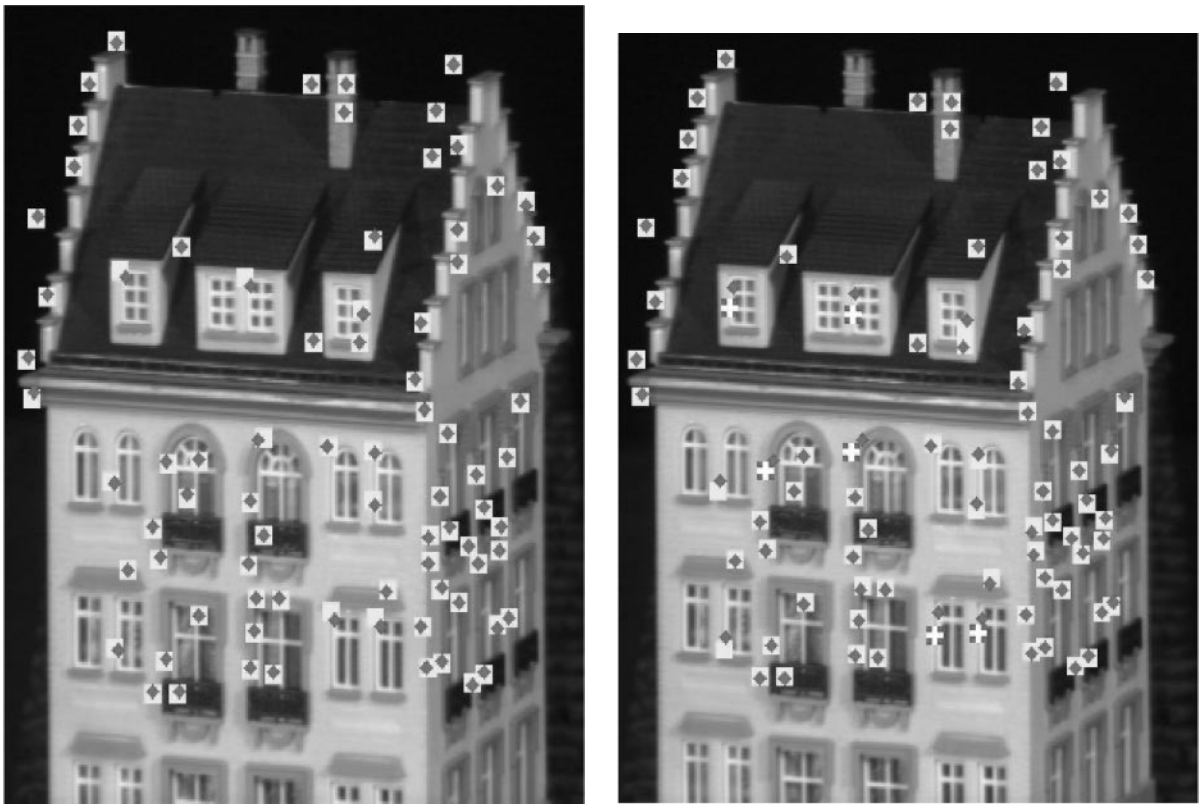


Figure 5. Two (cropped) frames from the hotel sequence showing a single sample reconstruction. Squares correspond to measurements with mask bit one (i.e. the measurement of that point in that frame is believed correct); a white cross on a dark background corresponds to a measurement with mask bit zero (i.e. the measurement of that point in that frame is believed incorrect); grey diamonds correspond to model predictions. The extent to which a diamond is centered within a square gives the extent to which a model prediction is supported by the data. In the right frame, at several locations the tracker has skipped to another feature for unknown reasons. In each case the reconstruction identifies the data point as being erroneous, and reprojects to a point in a significantly different position from the measurement reported by the tracker and lying where a correct measurement would be as seen by the position relative to the surface texture on the object.

given a data set. A particularly attractive feature is that special datasets require no additional analysis. For example, if every element in the image has the same colour, we expect the colour constancy sampler to produce a very wide spread of samples for the surface reflectance; similarly, if a structure from motion data set is obtained by a camera translating in its plane, the sampler will return a set of samples with substantial variance perpendicular to that plane without further ado. A second attractive feature is that both expectations and marginal probability distributions are easily available: to compute an expectation of a function, we average that function's value over the samples, and to compute a marginal, we drop irrelevant terms from the state of each sample.

Figure 7 illustrates the kind of information a sampler can produce for the structure from motion data; in particular, the sampler reflects the scatter of possible inferred values for a single point.

Figure 8 show a set of typical results a sampler can produce from real images for the colour constancy problem. The spatial model identifies edges correctly. Groups of samples drawn for the same surface reflectance under different lights intersect, as we expect. Furthermore, groups of samples drawn for different surface reflectances under the same light tend not to intersect, meaning that these surfaces are generally seen as different. The figure shows a rendering of samples under white light, to give some impression of the variation in descriptions that results.

5.2.2. Resampling to Incorporate New Information. Assume that we are engaged in colour constancy. We construct a representation of surface colour, and new

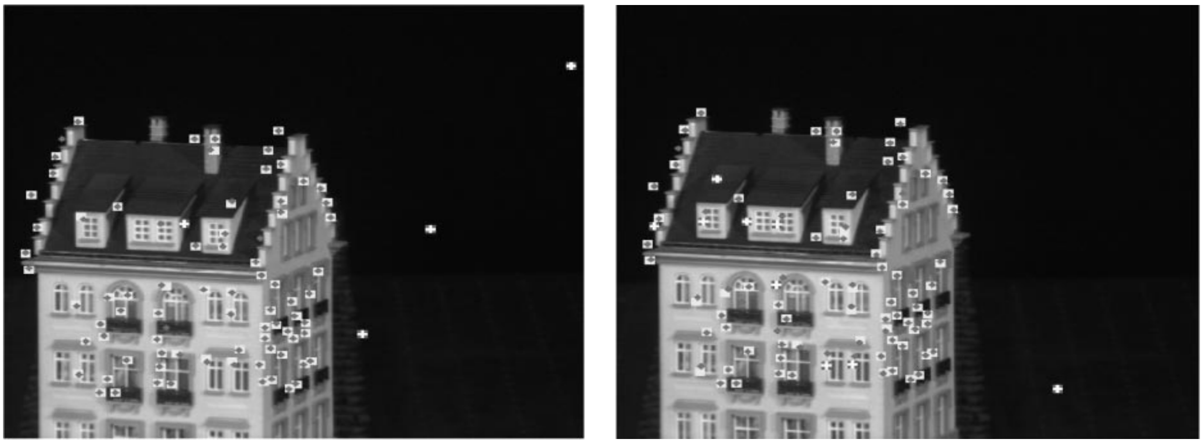


Figure 6. We perturb the hotel sequence by replacing 5% of the data points with draws from a uniform distribution in the image plane. The Bayesian method, started as in Section 5.4.1, easily discounts these noise points; the figure shows the same frames in the sequence as in Fig. 5, uncropped to show the noise but with a sample reconstruction indicated using the same notation as that figure.Squares correspond to measurements with mask bit one (i.e. the measurement of that point in that frame is believed correct); a white cross on a dark background corresponds to measurements with mask bit zero (i.e. the measurement of that point in that frame is believed incorrect); grey diamonds correspond to model predictions. The extent to which a diamond is centered within a square gives the extent to which a model prediction is supported by the data.

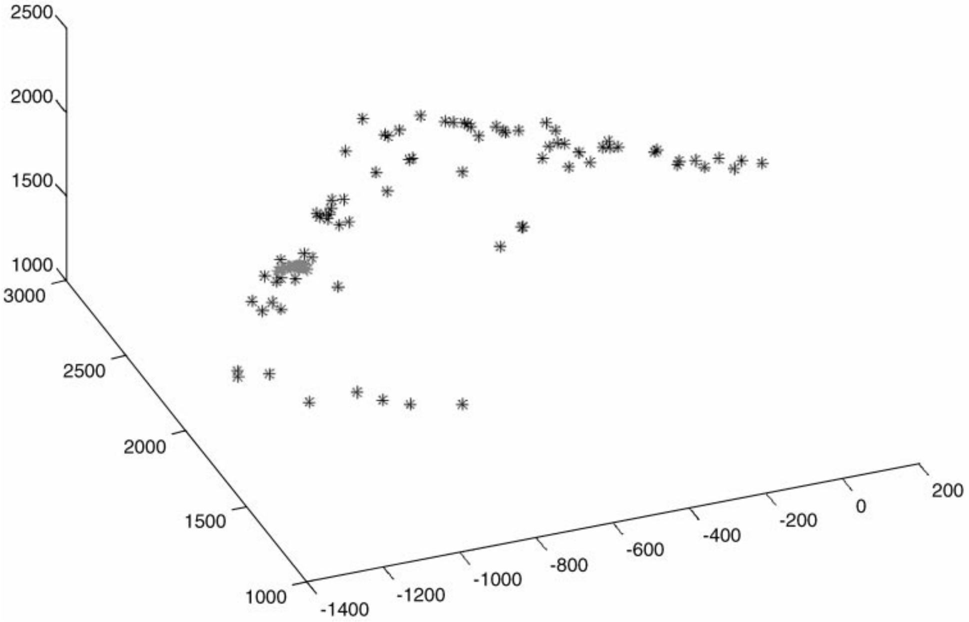


Figure 7. Black points show an overhead view of a single sample of the 3D reconstruction obtained using 40 frames of 80 points in the hotel sequence, rotated by hand to show the right-angled structure in the model indicating that the structure is qualitatively correct; the cloud of grey points are samples of the position of a single point, scaled by 1000 to show the (very small) uncertainty available in a single point measurement.

information arrives—what do we do? If the representation is probabilistic, the answer is (relatively) straightforward; we adjust our representation to convey the posterior incorporating this new information. For example, assume we have a sampled representation of the posterior for two distinct images. We are now told that a patch in one image is the same as a patch in another—this should have an impact on our interpretation of both images. The sampled representation is well suited to determining the effect of this information.

In particular, we have samples of

$$P(\sigma_a, \text{state a} \mid \text{image a})$$

and

$$P(\sigma_b, \text{state b} \mid \text{image b})$$

where we have suppressed the details of the rest of the state in the notation. We interpret “the same” to mean that each patch is a sample from a Gaussian distribution with some unknown mean α and a known standard deviation. We would like to obtain samples of

$$P(\alpha, \text{state a, state b} \mid \text{image a, image b})$$

(image a will be abbreviated as “im a”, etc.). Now we have that

$$P(\text{im a, im b} \mid \text{state a, state b, } \alpha)$$

is proportional to

$$\int \left(\begin{array}{c} P(\text{im a, state a} \mid \sigma_a) P(\sigma_a \mid \alpha) \times \\ P(\text{im b, state b} \mid \sigma_b) P(\sigma_b \mid \alpha) \end{array} \right) d\sigma_a \, d\sigma_b \pi(\alpha)$$

Now the term inside the integral is:

$$\frac{P(\text{state a}, \sigma_a, \text{image a})}{\pi(\sigma_a)} \times \frac{P(\text{state b}, \sigma_b, \text{image b})}{\pi(\sigma_b)} \times P(\sigma_b \mid \alpha) P(\sigma_a \mid \alpha)$$

We have two sets of samples, $\sum^a$ and $\sum^b$. We ensure that these samples are independent and identically distributed by shuffling them (to remove the correlations introduced by MCMC). This means that, for the conditional density for the i’th sample, we have $P(\sum_i^a \mid i) = P(\text{state a}, \sigma_a, \text{image a})$. Now we construct a new sampler, whose state is $\{i, j, \alpha\}$. We ensure this produces samples of the distribution

$$\Pi(i, j, \alpha) = \frac{\left(\begin{array}{c} P(\sigma_a(i) \mid \alpha) \times \\ P(\sigma_a(j) \mid \alpha) \pi(\alpha) \end{array} \right)}{\pi(\sigma_a(i)) \pi(\sigma_b(j))}$$

We now use the i’s and j’s as indexes to our previous set of samples. We can marginalise with respect to σ_a and σ_b by simply dropping their values from the sample. The result is a set of samples distributed according to the desired distribution:

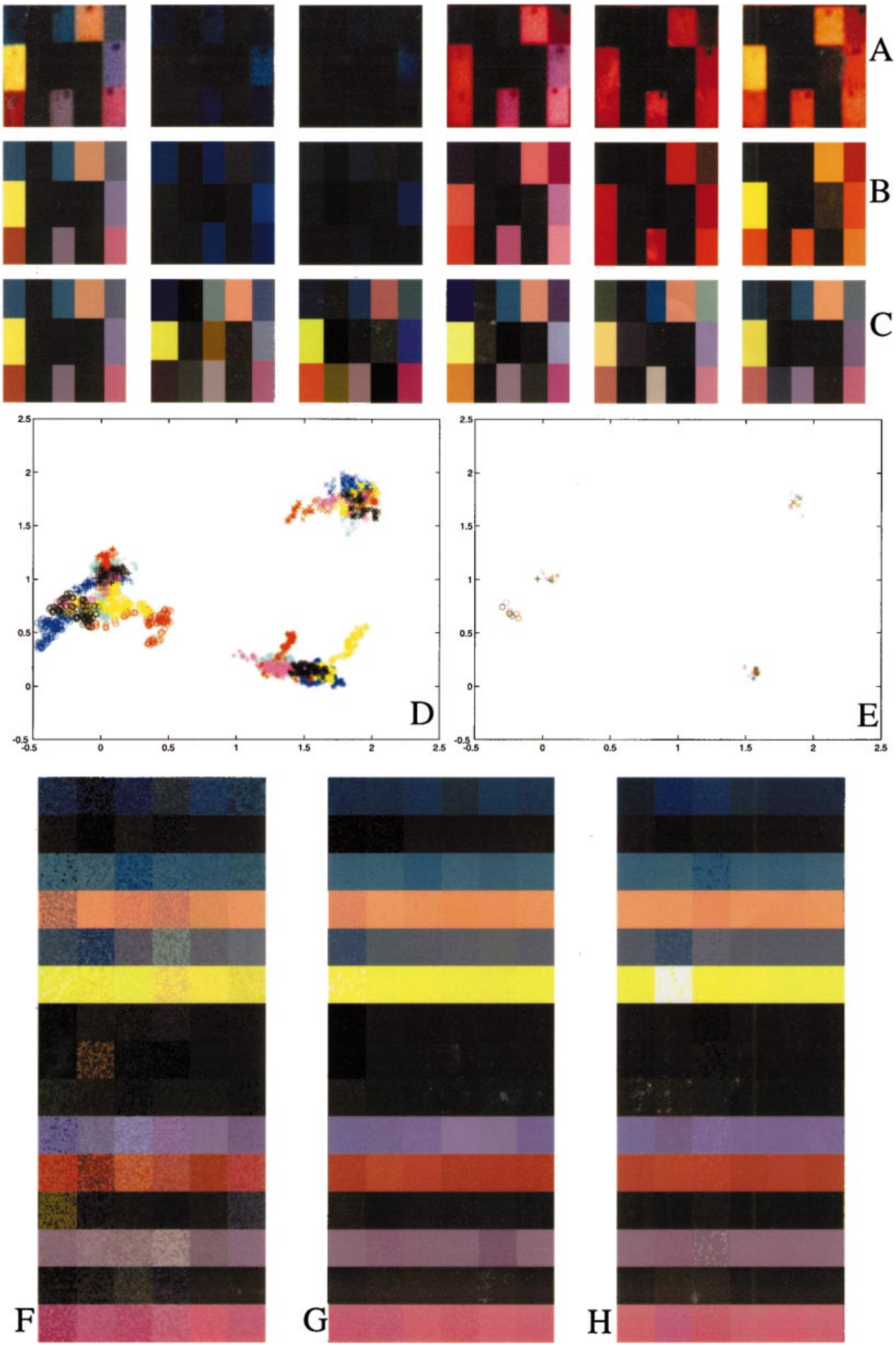
$$\int \left(\begin{array}{c} P(\text{im a, state a} \mid \sigma_a) P(\sigma_a \mid \alpha) \times \\ P(\text{im b, state b} \mid \sigma_b) P(\sigma_b \mid \alpha) \end{array} \right) d\sigma_a \, d\sigma_b \pi(\alpha)$$

Building a sampler that obtains samples of $\{i, j, \alpha\}$ space according to the desired distribution involves technical difficulties beyond the scope of this paper. The approach essentially chooses pairs consisting of a sample from the set for image a and a sample from the set for image b; these pairs are chosen with a frequency that is higher when the values inferred for a particular patch are similar. Of course, this trick extends to more images.

Figure 8 shows results obtained by assuming that a single surface patch in each of the six images is the same. Typically, a small number of sets of samples have a very much higher probability than all others, so that a sampled representation consists of a large number of copies of these samples, interspersed with one or two others. This results in very much reduced variance in the rendering of the patch that is known to be similar for the six images, because the error balls for this surface patch intersect in a relatively small region. However, this does *not* mean that the variance for the inferred reflectances for the other patches must be reduced. It *is* reduced (Fig. 8), but this is because the representations recovered for each separate input image (correctly) captures the possibility that each of the surface patches is the same. This is another important reality check that strongly suggests the sampled representation is trustworthy: the algorithm has been able to use information that one patch is the same in each image to obtain a representation that strongly suggests the other patches are the same, too.

5.2.3. Stability of the Recovered Representations. Reconstructions cannot be compared on the basis of accuracy, because “ground truth” is not available. However, we can demonstrate that sampled representations are stable under various perturbations of their input. In structure from motion, small errors in tracker response for some points could lead to significant perturbations of the reconstruction for all points, because the reconstructed point positions are not independent—they are coupled by the reconstructed camera configurations.

Small errors in tracker response actually occur: in the 40 frames of the hotel sequence that we used, six point measurements in nine frames are affected


A
B
C
2.5
2
1.5
1
0.5
0
-0.5
-0.5
0
0.5
1
1.5
2
2.5
D
2.5
2
1.5
1
0.5
0
-0.5
-0.5
0
0.5
1
1.5
2
2.5
E
F
G
H

by small tracker errors as shown in Fig. 5. These (very small) errors affect the reconstruction obtained using the factorisation method because the factorisation of a matrix is a function of all its entries (or equivalently, the reconstructed point positions are coupled by the reconstructed camera configurations).

To compare the stability of the methods, we now introduce larger tracker errors; a small percentage of data points, randomly selected, are replaced with draws from a uniform distribution on the image plane. If these points are included in the factorisation, the results are essentially meaningless. To provide a fair comparison, we use factorisations obtained using the method of Section 5.4.1 (these are the start points of our sampler). These reconstructions are guaranteed to ignore large error points but will ignore a significant percentage of the data.

In comparison, the sampler quickly accretes all points consistent with its model, and so gives significantly more stable measurements (cf Torr and Zisserman, 1998, which uses maximum likelihood to identify correspondences). Because the reconstruction is in some unknown scaled Euclidean frame, reconstructions are best compared by comparing angles subtended by corresponding triples of points, and by comparing distances between corresponding points scaled to minimize the errors. The sampled representation is significantly more stable under tracker errors and noise than a factorisation method (Figs. 9 and 10).

5.3. *Comparing Different Algorithms for Obtaining Covariance Estimates*

Probability distributions are devices for computing expectations. Computing an expectation is an integration problem; for high dimensional problems like those described here, “the curse of dimensionality” applies, and quadrature methods are not appropriate (e.g. the review of numerical integration methods in Evans and Swartz (2000). This leaves us with two possibilities: a random or quasi-random method, or an analytic approximation to the integral. Applying quasi-random methods to the problems described here appears to pose substantial technical difficulties; we refer the interested reader to Evans and Swartz (2000) and Traub and Werschulz (1999).

The analytic approximation most currently used in computer vision is based on Laplace’s method (described in Evans and Swartz (2000) and in the form we use it in Ripley (1996, p. 63); we shall call the approximation Laplace’s method in what follows). This approach models a unimodal posterior distribution with a normal distribution, whose mean is at the mode of the posterior and whose covariance matrix is the inverse of the Hessian of the posterior at the mode. In essence, the approximation notes that the main contribution to an expectation computed using a “peaky” probability distribution is at the mode; the contribution of the tails is estimated by the Hessian at the mode.

Figure 8. **A:** images of the same set of patches on a Mondrian of coloured paper patches, photographed under white, blue, purple, red, aqua and yellow light and scanned from Forsyth (1990), used as inputs to the sampler. **B:** renderings of typical representations obtained by the sampler, in each case shown under the coloured light inferred (so that in a successful result, the inferred representation looks like the image above it). Note the accuracy of the spatial model, and the robustness to image noise. **C:** renderings of typical representations under the same (white) light, so that a successful result implies similar renderings. **D:** The first two components of surface reflectance samples, plotted on the same axes for four different surfaces. Each sample is colour keyed to the image from which it was obtained; red samples for the red image, etc, with black corresponding to the white image. The circles show samples of the reflectance coefficients for the blue surface at the top left corner of the Mondrian; the stars for the yellow surface in the second row; the plusses show samples for the orange surface in the top row of the Mondrian and the crosses for the red surface in the bottom row. Each surface generates a “smear” of samples, which represent the uncertainty in the inferred surface reflectance, given the particular image input. There is an important consistency check in this data. Notice that the smear of samples corresponding to a particular surface in one image intersects, but is not the same as, the smear corresponding to that surface in another. This means that the representation envisages the possibility of their being the same, but does not commit to it. **E:** The first two components of surface reflectance samples, plotted on the same axes for four different surfaces. These come from the samples shown as D, resampled under the assumption that the blue surface in the top left hand corner of the Mondrian is the same for each image. We use the same representation and axes as in that figure. Notice that this single piece of information hugely reduces the ambiguity in the representation. **F:** Samples of reflectances returned for each patch on the Mondrian using the images shown as A (above), under each light, rendered under white light. There are four hundred samples per patch and per illuminant, each rendered as a small square; thus, a patch for which there is very little information shows a salt-and-pepper style texture. The rows show samples for the same patch under different illuminants; each column corresponds to an illuminant (in the order aqua, blue, purple, red, white and yellow). Notice the very substantial variation in appearance; white pixels denote samples which saturated. Notice also that for each patch there are samples that look similar. **G:** The samples obtained when all samples are resampled, assuming that the right (blue) patch is the same patch in each image. **H:** The samples obtained when all samples are resampled, assuming that the sixth (yellow) patch is the same patch in each image. Notice the substantial reduction in variance; while this constraint does not force the other patches to look the same, they do because they are in fact the same surface.

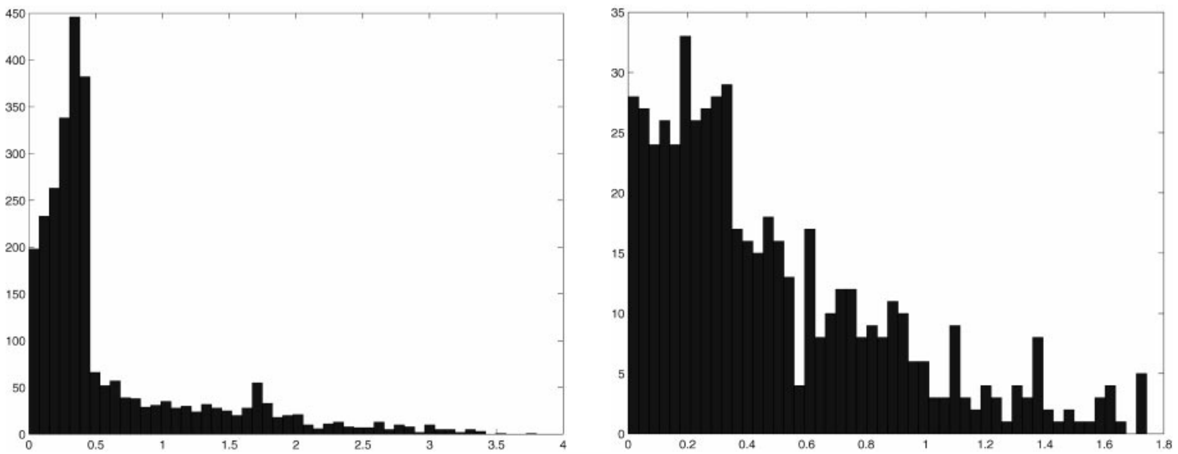


Figure 9. The factorisation method is relatively unstable under noise. We compare reconstructions obtained from the uncorrupted data set with reconstructions obtained when 5% of the entries in $\mathcal{D}$ are replaced with draws from a uniform distribution in the image plane; to represent the factorisation method fairly, we use the start points obtained using the algorithm of Section 5.4.1 (which masks off suspect measurements). *Left* shows a histogram of relative variations in distances between corresponding pairs of points and *right* shows a histogram of differences in angles subtended by corresponding triples of points. Note the scales—some interpoint distances are misestimated by a factor of 3, and some angles are out by $\pi/2$.

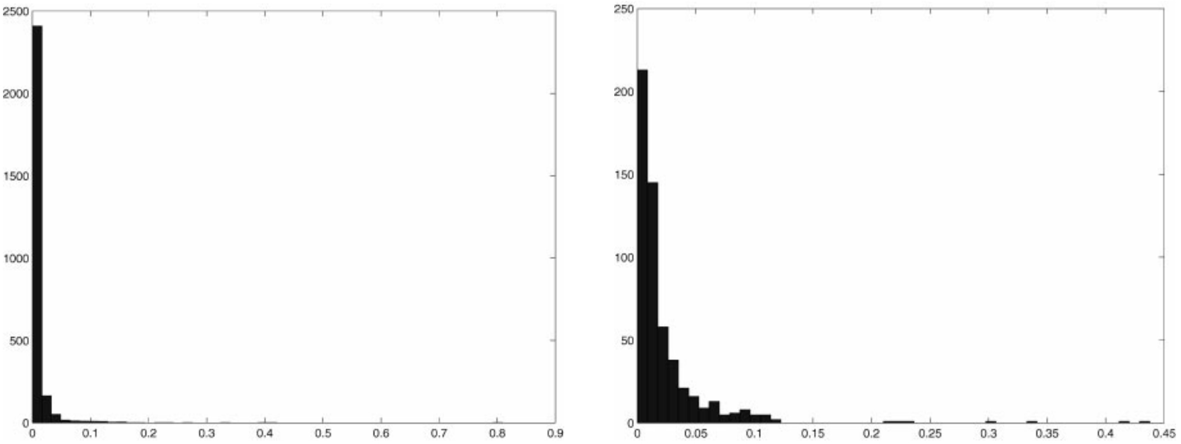


Figure 10. The Bayesian method is stable under noise. We compare reconstructions obtained from the uncorrupted data set with reconstructions obtained when 5% of the entries in $\mathcal{D}$ are replaced with draws from a uniform distribution in the image plane. *Left* shows a histogram of relative variations in distances between corresponding pairs of points and *right* shows a histogram of differences in angles subtended by corresponding triples of points. Note the significant increase in stability over the factorisation method; relative errors in distance are now of the order of 10% and angular errors are of the order of $\pi/40$.

Laplace's method is a natural linearisation, and has been used for estimates of covariance in the structure from motion literature (Morris and Kanade, 1998). However, as Fig. 11 indicates, the estimates it produces can differ substantially from the estimates produced by a sampler. As we have seen (Section 5.1), the sampler appears to mix acceptably, so this is not because the samples significantly understate the covariance (compare Fig. 11 with Fig. 4, which shows the order in which samples were drawn for the samples of Fig. 11). Instead, it is because Laplace's method approximates the probability density function poorly.

This is because the log of the posterior consists largely of terms of degree four. In such cases, the Hessian can be a significantly poor guide to the structure of the log-posterior a long way from the mode.

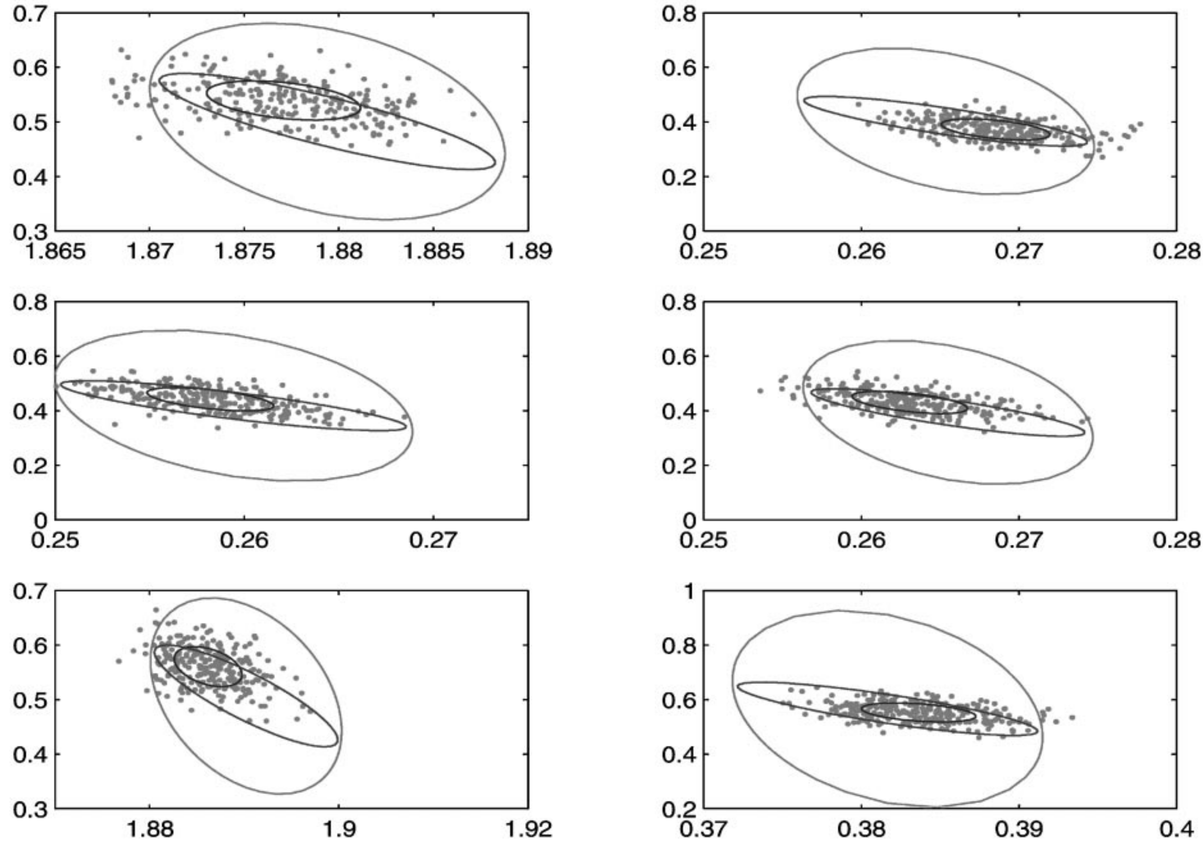


Figure 11. We compare the sampled representation of the posterior for the structure from motion problem with a representation obtained using an analytic approximation. Each of the six plots depict three different estimates of marginal posterior probabilities for point position in a plane parallel to the optical axis. (The points are the same points as in Fig. 4.) Samples are shown as a scatter plot. In each case, the one standard deviation ellipse for the covariance estimate obtained from Laplace's approximation is the largest of the three shown, and substantially overestimates covariance; its orientation is often misleading, too (it is plotted in light grey). In each case, the second largest ellipse is the one standard deviation ellipse obtained using Laplace's approximation and assuming that point and camera positions are independent; this is still an overestimate, but is a better estimate than that from Laplace's approximation (it is plotted in dark grey). Finally, the smallest ellipse in each case is obtained from the sample mean and covariance (it is plotted with the darkest grey). Laplace's approximation appears to significantly overestimate the covariance; this is almost certainly because the Hessian at the mode is a poor guide to the behaviour of the tails of the posterior for this problem.

In particular, it overestimates the weight of the tails and therefore overestimates the covariance. This is because it is a purely local estimate of the structure of the posterior—we cannot rely on the second derivative of a function at a point necessarily to convey helpful information about what the function is doing a long way away from that point. In comparison, each sample involves (at least!) a comparison of values of the posterior at that sample and at the previous sample, so that the samples are not relying on a local estimate for the structure of the posterior.

No really useful comparison is available for the case of colour constancy. All current colour constancy algorithms report either exact solutions, or minimum error solutions. Laplace's method should produce absurd covariance estimates, because the domain of integration is heavily truncated by the constraints of Section 3—the tails make no contribution, and it is unreasonable to expect a sensible approximation from the method.

5.4. *Speed*

Both samplers are relatively slow. Samples take longer to draw for the structure from motion problem (2000 samples for 40 views of 80 points in about a day on a

300 MHz Macintosh G3 system in compiled Matlab) than for the colour constancy problem (1000 samples in an hour in compiled Matlab on the same computer). While this is irritatingly slow, it does not disqualify the technology. In particular, it is important to keep in mind that cheaper technologies—the Laplace approximation estimate of covariance in Section 5.3 comes to mind—may offer significantly inaccurate representations. There are several possibilities for speedups:

- **An intelligent choice of start point:** there is no particular reason to start these samplers at a random start point and then wait for the gradient descent component of hybrid MCMC to find the mode. Instead, we can start the sampler at a decent estimate of the mode; we describe relevant methods below.
- **A faster mixing rate:** generally, the better a sampler mixes the fewer samples one needs to draw, because the samples increasingly mimic IID samples. It isn't clear how to build a truly fast-mixing sampler. The best strategy appears to be to use image data to structure the proposal distribution (as in Section 3 and Zhu et al., 2000), but there are no proofs that this leads to a fast-mixing sampler.
- **Lower per-sample cost:** it is unlikely that a decent representation of covariance will be available with fewer than 1000 samples. This means that each sample should be cheap to obtain. Current possibilities include: a faster integrator in the hybrid MCMC method (we used a symplectic Runge-Kutta-Nystrom method from Sanz-Serna and Calvo, 1994, with no effort to choose the fastest overall integrator); a grouping of the variables that allows an efficient Gibbs sampler (separating cameras and points leads to a standard form but a sampler that makes only minuscule changes of state for each sample, for the reason illustrated in Fig. 1); and fitting a Gaussian at each sample and using this Gaussian to propose a new state.[1]

5.4.1. Starting the SFM Sampler. The sampler's state is given by $(\mathcal{U}, \mathcal{V}, \mathcal{M})$. We show examples for $(m, n) = (40, 80)$ and $(m, n) = (24, 100)$. This means the domain of the sampler is then 2^{3200} (resp. 2^{2400}) copies of $\Re^{640}$ (resp. $\Re^{592}$). The relations between the discrete and the continuous variables are complex; for small errors, a sampler started at a random point burns in relatively quickly, but for large errors, the burn in can be very slow.

The values of $\mathcal{U}$ and $\mathcal{V}$ depend strongly on $\mathcal{M}$. If $\mathcal{M}$ has a 1 in a position corresponding to a significant tracker error, then that error can strongly affect the values of $\mathcal{U}$ and $\mathcal{V}$. This effect slows down the convergence of the sampler, because incorrect values of the continuous parameters mean that many data points lie a long way from the values predicted by the model, so that there is little distinction between points that correspond to the model and points that do not.

We start the sampler at a fair initial estimate of the mode. We obtain an initial value for the mask $\mathcal{M}^a$ by sampling an independent distribution on the bits that tends to deemphasize points which are distant from corresponding points in the previous and next frames. In particular, the i, j'th bit of $\mathcal{M}^a$ is 0 with probability

$$\frac{1 - \exp\left(\frac{-\Delta_{ij}}{\sigma_w}\right)}{1 + \exp\left(\frac{-\Delta_{ij}}{\sigma_w}\right)}$$

where $\Delta_{ij} = (d_{i,j} - d_{i+1,j})^2 + (d_{i+m,j} - d_{i+m+1,j})^2 + (d_{i,j} - d_{i-1,j})^2 + (d_{i+m,j} - d_{i+m-1,j})^2$. Since this is a problem where the quantity of data swamps the number of parameters in the model, the choice of σ_w is fairly unimportant; the main issue is to choose the value to be small enough that large tracker errors are masked almost certainly.

The $\mathcal{U}^a$ and $\mathcal{V}^a$ that maximise

$$\sum_{ij} \left\{ \left(d_{ij} - \sum_k u^a_{ik} v^a_{kj} \right)^2 m^a_{ij} \right\}$$

are then obtained by a sweep algorithm which fixes $\mathcal{U}$ (resp. $\mathcal{V}$) and solves the linear system for $\mathcal{V}$ (resp. $\mathcal{U}$), and then swaps variables; the sweeps continue until convergence (which is guaranteed). We now compute an affine transformation $\mathcal{A}$ such that $\mathbf{C}^T(\mathcal{U}^a\mathcal{A}, \mathcal{A}^{-1}\mathcal{V}^a)\mathbf{C}(\mathcal{U}^a\ \mathcal{A},\ \mathcal{A}^{-1}\mathcal{V}^a)$ is minimised; then $\mathcal{U}^s = \mathcal{U}^a\mathcal{A}$ and $\mathcal{V}^s = \mathcal{A}^{-1}\mathcal{V}^a$. We now draw a sample from the full conditional on each bit in the mask matrix, given $\mathcal{U}^s$ and $\mathcal{V}^s$ to obtain $\mathcal{M}^s$ The start state is then $(\mathcal{U}^s,\ \mathcal{V}^s,\ \mathcal{M}^s)$.

5.4.2. Starting the Colour Constancy Sampler. The sampler converges if started from a random sample from the prior, but this is slow and unnecessarily inefficient. A good guess at edge positions follows by choosing a set of edges at maxima of the edge proposal distributions, censored to ensure the hardcore model applies. Similarly, a start point for the light position follows by choosing the maximum likelihood position from the proposal distribution; once the specular position is known, an estimate of illuminant colour follows. Finally, for each patch we obtain a reflectance estimate from the average colour within the patch and the illuminant colour. This yields a start point from which the sampler converges relatively quickly.

6. Discussion—Ups and Downs of Sampling Methods

Good samplers are fast, burn in quickly, and mix well. It can be proven that some samplers are good (at least in theory) and some are obviously bad; most are merely mysterious as to their behaviour. It is possible to build samplers that yield representations that pass a wide range of sanity checks, and some of these are fairly fast. This is probably the best that can be hoped for in the near future.

6.1. Points in Favour of Using Sampled Representations

There are several points in favour of using sampled representations: The strongest is the **simple management of uncertainty** that comes with such methods. Once samples are available, managing information is simple. Computing expectations and marginalization, both useful activities, are particularly easy. Incorporating new information is, *in principle*, simple. The output of a properly built sampler is an excellent guide to the inferences which can be drawn and to the ambiguities in a dataset. For example, in Fig. 7, we show uncertainty in the position of a single point in space (determined by a structure from motion method) as a result of image noise. No independence assumptions are required to obtain this information; furthermore, we are not required to use specialised methods when the camera motion is degenerate—if, for example, the camera translates within a plane, the effect will appear in scatter plots that vary widely along the axis perpendicular to the plane.

The main benefit that results is simple **information integration**. Building vision systems on a reasonable scale requires cue integration; for example, what happens if colour reports a region is blue, and shape says it's a fire engine? this contradiction can only be resolved with some understanding of the reliability of the reports. A properly-built Bayesian model incorporates all available information, and is particularly attractive when natural likelihood and prior models are available (e.g. examples in Sections 2 and 3). In principle, sampling can work for arbitrary posteriors.

Another feature of sampled methods is that they can handle **complex spatial models**. The main difficulty with such models is domains with complicated topologies. For example, it is simple to deal with a domain which consists of many components of different dimension (Green, 1995). This means that a spatial model can be part of the posterior. For example, in Section 3, we model the layout of a Mondrian as a grid of rectangles, where neither the position nor the number of the horizontal and vertical edges of the grid are known. Instead, these are inferred from data. This offers the prospect of unifying information about coherence, spatial layout and model appearance by performing segmentation with explicit spatial models. Sampling methods are a standard approach to performing inference using spatial models (Geyer, 1999; Moller, 1999).

6.2. The Problems with Samplers

While samplers are in principle generic, in practice building a good sampler requires a significant degree of skill. **The number of samples required** can be very large. Vision problems typically consist of large numbers of discrete and continuous variables. If a posterior is a complicated function of a high dimensional space, with many important modes, an extremely large number of samples may be required to support any useful representation (either as samples, or as a mixture model or some other simplified parametric model fitted to samples). However, for most well phrased vision problems, we expect to see a small number of quite tight modes in the posterior, suggesting that the relevant portion of the posterior could be represented by manageable numbers of samples; furthermore, an accurate representation of tails is a less significant need than a reasonable description of the modes.

Samplers are currently relatively **slow**. However, it is possible to build samplers that are fast enough that useful solutions to real vision problems can be obtained in reasonable amounts of time. Generally, the prospect of understanding how to build *better* systems precedes understanding how to build *faster* systems.

Sampled representations have a claim to **universality**. Any conceivable representation scheme appears to rest on the presence of samples. For example, one might wish to approximate a posterior as a mixture model. To do so, one can either fit the model to a set of samples, or compute various integrals representing the error; but good numerical integrators in high dimensions are based on sampling methods of one form or another. This suggests that, unless a problem can be persuaded to take a series of manageable parametric forms *for which deterministic algorithms for computing fits are available*, one is stuck with the difficulties that come along with sampling methods.

Vision problems often have a form that is **well adapted** to sampling methods. In particular, there is usually a preponderance of evidence, meaning that the posterior should have few, large, well-isolated peaks, *whose location can be estimated*. Furthermore, it is commonly the case that computer vision algorithms can compute values for some variables given others are known. The Metropolis-Hastings algorithm gives a framework within which such algorithms can be integrated easily, to produce a series of hypotheses *with meaningful semantics*.

Samplers are poorly adapted to problems that lead to **large domains which have essentially uniform probability**. This might occur, for example, in an MRF model where there may be a very large number of states with essentially the same, near-maximal, posterior probability, because each is a small number of label-flips away from the extremum. The difficulty is not the sampler, but the representation it produces. It is quite easy to set up examples that require very large numbers of samples to represent these regions, particularly if the dimension of the domain is large. A fair case can be made that such problems should properly be reparametrised (perhaps by imposing a parametric form) *whatever strategy is to be adopted for addressing them*: firstly, because large domains of essentially uniform probability suggest that some problem parameters don't have any significant effect on the outcome; secondly, because estimates of the mode will be extremely unstable; and thirdly, because any estimator of an expectation for such a problem must have high variance.

When can samples be trusted? Typically, the first k samples must be discarded to allow the sampler to "burn in". The rest represent the posterior; but what is k?. The usual approach is to start different sequences at different points, and then confirm that they give comparable answers (e.g. Gelman and Rubin, 1993; Geweke, 1992; Roberts, 1992). Another approach is to prove that the proposal process has rapid mixing properties (which is extremely difficult, e.g. Jerrum and Sinclair, 1996). Rapid mixing is desirable, because the faster the sampler mixes, the lower the variance of expectations estimated using samples (Geyer, 1999). The only mechanism available for many practical problems is to structure one's experimental work to give checks on the behaviour of the sampler. For example, in the work on structure from motion the sampler was able to identify bad measurements and gave stable reconstructions (Section 5.2); similarly, in the work on colour constancy the resampling algorithm correctly reduced the variance in the inferred colour of other patches when informed that some patches had the same colour (Section 5.2).

6.3. *Reasons to be Cheerful*

Interesting vision problems are well-behaved enough to make samplers quite practical tools. Firstly, in most vision problems there is an overwhelming quantity of data compared to the number of parameters being studied; as a result, it is usual to expect that the posterior might have a very small number of quite well-peaked modes, so that exploration of the domain of the sampler can be restricted to small subsets. Secondly, there is a substantial body of algorithms that make good estimates at the position of these modes (e.g. derivative filters estimating the position of edges; factorisation estimating structure and motion; etc.), so that a sampler can be started at a good state. Finally, many vision problems display a kind of conditional independence property that allows a large problem to be decomposed into a sampling/resampling problem (e.g. Section 3, and Ioffe and Forsyth, 1999).

Acknowledgments

This research was supported in part by a grant from Adobe systems, in part by an NSF Fellowship to SI, in part by an NSF Digital Library award (NSF IIS 9817353). The hotel sequence appears courtesy of the Modeling by Videotaping group in the Robotics Institute, Carnegie Mellon University. Thanks to Stuart Russell for pointing out the significance of MCMC as an inference technique.

Note

1. We are indebted to Andrew Zisserman for this suggestion.

References

Amit, Y., Grenander, U., and Piccioni, M. 1991. Structural image restoration through deformable templates. *J. Am. Statist. Ass.*, 86:376–387.

Beardsley, P.A., Zisserman, A.P., and Murray, D.W. 1997. Sequential updating of projective and affine structure from motion. *Int. J. Computer Vision*, 23(3):235–259.

Besag, J., Green, P., Higdon, D., and Mengersen, K. 1995. Bayesian computation and stochastic systems. *Statistical Science*, 10(1): 3–41.

Binford, T.O. and Levitt, T.S. 1994. Model-based recognition of objects in complex scenes. In *Image Understanding Workshop*, pp. 149–155.

Blake, A. and Isard, M. 1998. Condensation—conditional density propagation for visual tracking. *Int. J. Computer Vision*, 29(1): 5–28.

Blake, A. and Zisserman, A. 1987. *Visual Reconstruction*. Cambridge, MA: MIT Press.

Brainard, D.H. and Freeman, W.T. 1997. Bayesian colour constancy. *J. Opt. Soc. Am.-A*, 14:1393–1411.

Buchsbaum, G. 1980. A spatial processor model for object colour perception. *J. Franklin Inst.*, 310:1–26.

Carlin, B.P. and Louis, T.A. 1996. *Bayes and Empirical Bayes Methods for Data Analysis*. Chapman and Hall.

Carpenter, J., Clifford, P., and Fearnhead, P. 1999. Improved particle filter for non-linear problems. *IEEE Proc. Radar, Sonar and Navigation*, 146(1):2–7.

Chou, P.B. and Brown, C.M. 1990. The theory and practice of bayesian image labeling. *Int. J. Computer Vision*, 4(3):185–210.

Collins, N.E., Englese, R.W., and Golden, B.L. 1988. Simulated annealing—an annotated bibliography. Technical report, University of Maryland at College Park, College of Business and Management.

Costeira, J.P. and Kanade, T. 1998. A multibody factorisation method for independently moving objects. *Int. J. Computer Vision*, 29(3):159–180.

Debevec, P.E., Taylor, C.J., and Malik, J. 1996. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *SIGGRAPH '96*, pp. 11–20.

Dellaert, F., Seitz, S., Thorpe, C., and Thrun, S. 2000. Structure from motion without correspondence. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 557–564.

Doucet, A., De Freitas, N., and Gordon, N. 2001. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York.

Duane, S., Kennedy, A.D., Pendleton, B.J., and Roweth, D. 1987. Hybrid monte carlo. *Physics Letters B*, 195:216–222.

Evans, M. and Swartz, T. 2000. *Approximating Integrals via Monte Carlo and Deterministic Methods*. Oxford University Press: New York.

Faugeras, O.D. and Robert, L. 1996. What can two images tell us about a third one? *Int. J. Computer Vision*, 18(1):5–19.

Faugeras, O., Robert, L., Laveau, S., Csurka, G., Zeller, C., Gauclin, C., and Zoghlami, I. 1998. 3d reconstruction of urban scenes from image sequences. *Computer Vision and Image Understanding*, 69(3):292–309.

Finlayson, G. Colour in perspective. 1996. *IEEE T. Pattern Analysis and Machine Intelligence*, 18:1034–1038.

Fischler, M.A. and Bolles, R.C. 1981. Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography. *Comm. ACM*, 24:381–395.

Forsyth, D.A. 1990. A novel algorithm for colour constancy. *Int. J. Computer Vision*, 5:5–36.

Funt, B.V., Barnard, K., and Martin, L. 1998. Is machine colour constancy good enough? In *ECCV*, pp. 445–459.

Gamerman, D. 1997. *Markov Chain Monte Carlo*. Chapman-Hall: New York.

Gelman, A. and Rubin, D.B. 1993. Inference from iterative simulation using multiple sequences. *Statistical Science*, 7: 457–511.

Gelman, A., Carlin, J.B., Stern, H.S., and Rubin, D.B. 1995. *Bayesian Data Analysis*. Chapman and Hall: London.

Geman, S. and Geman, D. 1984. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE T. Pattern Analysis and Machine Intelligence*, 6:721–741.

Geman, S. and Graffigne, C. 1986. Markov random field image models and their application to computer vision. In *Proc. Int. Congress of Math.*

Geweke, J. 1992. Evaluating the accuracy of sampling based approaches to the calculation of posterior moments. In *Bayesian Statistics 4*, J.M. Bernardo, J. Berger, A.P. Dawid, and A.F.M. Smith (Eds.). Oxford: Clarendon Press.

Geyer, C. 1999. Likelihood inference for spatial point processes. In *Stochastic Geometry: Likelihood and Computation*. O.E. Barndorff-Nielsen, W.S. Kendall, and M.N.W. van Lieshout (Eds.), Chapman and Hall: Boca Raton.

Gilks, W.R. and Roberts, G.O. 1996. Strategies for improving mcmc. In *Markov Chain Monte Carlo in Practice*, W.R. Gilks, S. Richardson, and D.J. Spiegelhalter (Eds.), Chapman and Hall: New York.

Gilks, W.R., Richardson, S., and Spiegelhalter, D.J. 1996. Introducing Markov Chain Monte Carlo. In *Markov Chain Monte Carlo in Practice*, W.R. Gilks, S. Richardson, and D.J. Spiegelhalter (Eds.), Chapman and Hall: New York.

Gilks, W.R., Richardson, S., and Spiegelhalter, D.J. 1996. Introduction to markov chain monte carlo. In *Markov Chain Monte Carlo in Practice*, W.R. Gilks, S.Richardson, and D.J. Spiegelhalter (Eds.), Chapman and Hall: New York.

Gilks, W.R., Richardson, S., and Spiegelhalter, D.J. (Eds.). 1996. *Markov Chain Monte Carlo in Practice*. Chapman and Hall: New York.

Golden, B.L. and Skiscim, C.C. 1986. Using simulated annealing to solve routing and location problems. *Naval Res. Log. Quart.*, 33:261–279.

Van, Gool L. and Zisserman, A.P. 1997. Automatic 3d model building from video sequences. *European Transactions on Telecommunications*, 8(4):369–378.

Green, P.J. 1995. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4):711–732.

Green, P.J. 1996. Mcmc in image analysis. In *Markov chain Monte Carlo in practice*, W.R. Gilks, S. Richardson, and D.J. Spiegelhalter (Eds.), Chapman and Hall, New York, pp. 381–400.

Grenander, U. 1983. Tutorial in pattern theory. Technical report, Brown University, Providence, Rhode Island.

Grenander, Ulf. 1993. *General Pattern Theory*. Oxford University Press: New York.

Hartley, R. and Zisserman, A. 2000. *Multiple View Geometry*. Cambridge University Press: New York.

Huang, T., Koller, D., Malik, J., Ogasawara, G., Rao, B., Russell, S., and Weber, J. 1994. Automatic symbolic traffic scene analysis using belief networks. In *AAAI*, pp. 966–972.

Ioffe, S. and Forsyth, D.A. 1999. Finding people by sampling. In *Int. Conf. on Computer Vision*, pp. 1092–1097.

Jacobs, D. 1997. Linear fitting with missing data: Applications to structure-from-motion and to characterizing intensity images. In *IEEE Conf. on Computer Vision and Pattern Recognition*.

Jerrum, Mark and Sinclair, Alistair. 1996. The Markov chain Monte Carlo method: An approach to approximate counting and integration. In *Approximation Algorithms for NP-Hard Problems*, D.S. Hochbaum (Ed.), PWS Publishing: Boston.

Dubuisson Jolly, M.-P., Lakshmanan, S., and Jain, A.K. 1996. Vehicle segmentation and classification using deformable templates. *IEEE T. Pattern Analysis and Machine Intelligence*, 18(3):293–308.

Kanazawa, K., Koller, D., and Russell, S. 1995. Stochastic simulation algorithms for dynamic probabilistic networks. In *Proc Uncertainty in AI*.

Kitagawa, G. 1987. Non-gaussian state space modelling of non-stationary time series with discussion. *J. Am. Stat. Assoc.*, 82: 1032–1063.

Land, E.H. and McCann, J.J. 1971. Lightness and retinex theory. *J. Opt. Soc. Am.*, 61(1):1–11.

Lee, H.C. 1986. Method for computing the scene-illuminant chromaticity from specular highlights. *J. Opt. Soc. Am.-A*, 3:1694–1699.

Li, S.Z. 1995. *Markov Random Field Modeling in Computer Vision*. Springer-Verlag: New York.

Maloney, L.T. and Wandell, B.A. 1986. A computational model of color constancy. *J. Opt. Soc. Am.*, 1:29–33.

Marimont, D.H. and Wandell, B.A. 1992. Linear models of surface and illuminant spectra. *J. Opt. Soc. Am.-A*, 9:1905–1913.

Maybank, S.J. and Sturm, P.F. 1999. Minimum description length and the inference of scene structure from images. In *IEE Colloquium on Applied Statistical Pattern Recognition*, pp. 9–16.

McLachlan, G.J. and Krishnan, T. 1996. *The EM Algorithm and Extensions*. John Wiley and Sons: New York.

Moller, J. 1999. Markov chain Monte Carlo and spatial point processes. In *Stochastic Geometry: Likelihood and Computation*, O.E. Barndorff-Nielsen, W.S. Kendall, and M.N.W. van Lieshout (Eds.), Chapman and Hall: Boca Raton.

Morris, D.D. and Kanade, T. 1998. A unified factorization algorithm for points, line segments and planes with uncertainty models. In *Int. Conf. on Computer Vision*, pp. 696–702.

Mumford, D. and Shah, J. 1989. Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math.*, 42:577–684.

Neal, R.M. 1993. Probabilistic inference using markov chain monte carlo methods. Computer science tech report crg-tr-93-1, University of Toronto.

Noble, J.A. and Mundy, J. 1993. Toward template-based tolerancing from a bayesian viewpoint. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 246–252.

Pavlovic, V., Frey, B.J., and Huang, T.S. 1999. Time-series classification using mixed-state dynamic bayesian networks. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 609–612.

Pavlovic, V., Rehg, J.M., Cham, Tat-Jen, and Murphy, K.P. 1999. A dynamic bayesian network approach to figure tracking using learned dynamic models. In *Int. Conf. on Computer Vision*, pp. 94–101.

Phillips, D.B. and Smith, A.F.M. 1996. Bayesian model comparison via jump diffusion. In *Markov Chain Monte Carlo in Practice*, W.R. Gilks, S. Richardson, and D.J. Spiegelhalter (Eds.), Chapman and Hall.

Poelman, C. 1993. The paraperspective and projective factorisation method for recovering shape and motion. Cmu cs-93-219, Carnegie-Mellon University.

Richardson, S. and Green, P.J. 1987. On bayesian analysis of mixtures with an unknown number of components. *Proc. Roy. Stat. Soc. B*, 59:731–792.

Ripley, B. 1987. *Stochastic Simulation*. Wiley.

Ripley, B.D. 1996. *Pattern Recognition and Neural Networks*. Cambridge University Press.

Roberts, G.O. 1992. Convergence diagnostics of the gibbs sampler. In *Bayesian Statistics 4*, J.M. Bernardo, J. Berger, A.P. Dawid, and A.F.M. Smith (Eds.), Oxford: Clarendon Press.

Roberts, G.O. 1996. Markov chain concepts related to sampling algorithms. In *Markov chain Monte Carlo in Practice*, W.R. Gilks, S. Richardson, and D.J. Spiegelhalter (Eds.), Chapman and Hall: New York.

Rousseeuw, P.J. 1987. *Robust Regression and Outlier Detection*. Wiley: New York.

Sanz-Serna, J.M. and Calvo, M.P. 1994. *Numerical Hamiltonian Problems*. Chapman and Hall: New York.

Sarkar, S. and Boyer, K.L. 1992. Perceptual organization using bayesian networks. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 251–256.

Sarkar, S. and Boyer, K.L. 1994. Automated design of bayesian perceptual inference networks. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 98–103.

Sullivan, J., Blake, A., Isard, M., and MacCormick, J. 1999. Object localization by bayesian correlation. In *Int. Conf. on Computer Vision*, pp. 1068–1075.

Tierney, L. 1996. Introduction to general state-space markov chain theory. In *Markov Chain Monte Carlo in Practice*, W.R. Gilks, S. Richardson, and D.J. Spiegelhalter (Eds.), Chapman and Hall: New York.

Tomasi, C. and Kanade, T. 1992. Shape and motion from image streams under orthography: A factorization method. *Int. J. of Comp. Vision*, 9(2):137–154.

Torr, P.H.S. and Murray, D.W. 1997. The development and comparison of robust methods for estimating the fundamental matrix. *Int. J. Computer Vision*, 24:271–300.

Torr, P. and Zisserman, A. 1998. Robust computation and parametrization of multiple view relations. In *Int. Conf. on Computer Vision*, pp. 485–491.

Traub, J.F. and Werschulz, A. 1999. *Complexity and Information*. Cambridge University Press.

Triggs, B. 1995. Factorization methods for projective structure and motion. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 845–851.

Yuille, A.L. and Coughlan, J. 1999. High-level and generic models for visual search: When does high level knowledge help? In *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 631–637.

Zhu, S.C., Wu, Y., and Mumford, D. 1998. Filters, random fields and maximum entropy (frame): Towards a unified theory for texture modelling. *Int. J. Computer Vision*, 27:107–126.

Zhu, S.C. 1998. Stochastic computation of medial axis in markov random fields. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 72–80.

Zhu, S.C., Zhang, R., and Tu, Z. 2000. Integrating bottom-up/top-down for object recognition by data driven markov chain monte carlo. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 738–745.